Machine learning for human cancer research

Jong, C.

2006

Machine Learning for Human Cancer Research

Kees Jong

Typeset with LaTeX2$_\varepsilon$
Printed by Universal Press, The Netherlands

THOMAS STIELTJES INSTITUTE
FOR MATHEMATICS

VRIJE UNIVERSITEIT

Machine learning for human cancer research

ACADEMISCH PROEFSCHRIFT

ter verkrijging van de graad Doctor aan
de Vrije Universiteit Amsterdam,
op gezag van de rector magnificus
prof.dr. T. Sminia,
in het openbaar te verdedigen
ten overstaan van de promotiecommissie
van de faculteit der Exacte Wetenschappen
op donderdag 1 juni 2006 om 10.45 uur
in het auditorium van de universiteit,
De Boelelaan 1105

door

Cornelis Jong

geboren te Schagen

promotoren:	prof.dr. A.W. van der Vaart
	prof.dr. A.E. Eiben
copromotor:	dr. E. Marchiori

Contents

Chapter 1

Introduction

Cancer is a major public health problem. Currently, one in four deaths in the United States is due to cancer [25]. A total of 1,372,910 new cancer cases and 570,280 deaths are expected in the United States in 2005. When deaths are aggregated by age, cancer has surpassed heart disease as the leading cause of death for persons younger than 85 since 1999 [25]. The three most common cancer sites for men are lung and bronchus, colon and rectum, and prostate. The most common sites for women are breast and colorectal [25]. According to the "Centraal Bureau voor de Statistiek" (CBS, www.cbs.nl) cancer will be the leading cause of death in the Netherlands by the year 2010. These numbers justify a strong interest for this subject from researchers.

There are a few widely accepted causes of cancer, for instance tobacco [12]. Although many forms of cancer are sporadic, there are some cancers with a hereditary component, such as breast cancer, which has been related to genes named BRCA1 and BRCA2 (BRCA stands for breast cancer) [19].

Cancer results from molecular events that change the properties of human cells. In cancer cells the normal control systems that prevent cell overgrowth and the invasion of other tissues are disabled.

The abnormalities in cancer cells usually result from aberrations in genes that regulate cell division. Over time more genes become aberrated. This is often because the genes that make the proteins that normally repair DNA damage are themselves not functioning normally, because they are also aberrated. Consequently, aberrations begin to increase in the cell, causing further abnormalities in that cell. Some of these aberrated cells die, but other alterations may give the abnormal cell a selective advantage that allows it to multiply much more rapidly than normal cells.

Cancer research	Machine learning
Diagnosis	Classification
Tumor class discovery	Clustering
Biomarker identification	Feature selection

Table 1.1: Schema of machine learning in cancer research.

1.1 Motivation

Recent technological developments in molecular biology provide the opportunity to perform large scale measurements at the various "information levels" in the human cell: chromosomal DNA copy numbers, gene expression and protein expression. The analysis, presentation and interpretation of the vast amount of numerical data are a major challenge.

A particular type of experiment, called array CGH, provides data that should allow us to estimate chromosomal copynumber changes. In this thesis the problem of the experimental noise generated by this type of experiment is addressed. Furthermore a method is presented that allows us to perform a cross platform analysis of this type of data. This is interesting, since comparing many different studies may provide interesting new insights.

A second type of experiment is micro-array gene expression experiments that provide the opportunity to compare expression profiles of healthy persons and those who suffer from cancer. It is interesting to test the predictive power of these profiles, as a good predictor could be used for diagnosis. A novel clustering method based on the classification method called support vector machine (SVM) is presented to do so.

Mass spectra proteomics experiments give insight in the amount of small peptides and proteins present in cancer and healthy samples. Since proteins are responsible for much of the behavior of human cells, such experiments may provide sufficient information to perform a successful diagnosis. Furthermore an understanding of which proteins are important in certain types of cancer is of fundamental relevance, since it may lead to development of novel medicines. For this task a novel feature selection method based on evolutionary computation and SVMs is presented.

1.2 Machine learning in cancer research

Some cancer research questions can be associated with machine learning tasks. Table 1.1 shows some instances of this observation that are addressed in this thesis. In the following sections of this introduction we discuss the associations in table 1.1 in more detail and briefly discuss the laboratory and machine learning techniques that are considered in this thesis.

Classification for diagnosis

A classifier is a function from the set of possible examples to the set of possible classes. Typically the function has some parameters which need to be estimated. This is done by a learning algorithm, by using a number of examples with desired class given.

Diagnosis can be described as the process of identifying a disease by the symptoms. We can use a classifier for diagnosis of a single disease if we take the set of possible classes to be healthy or disease. The examples have values for a number of variables that are thought to be sufficient for diagnosis. What makes a good classifier depends on the application. In traditional machine learning we are looking for probably approximately correct (PAC) classifiers. That is, the probability that we make more than a certain number of misclassifications should be small.

In diagnostics a classification system may be used to make a "first selection". The patients who are almost certainly healthy should not be investigated further. To be more specific, if a patient is tested to indicate the presence of cancer, the result is "positive" if cancer is present, and "negative" if it is not. Suppose we have a set of patients for which we know whether they have cancer or not. For the applied test we can determine the number of patients correctly diagnosed to have cancer (true positive, tp), falsely diagnosed to have cancer (false positive, fp), correctly diagnosed not to have cancer (true negative, tn) and finally falsely diagnosed not to have cancer (false negative, fn). For a "first selection" a small number of false negatives is more important than a small number of false positives. In other words sensitivity (tp / (tp + fn)) is more important than specificity (tn / (tn + fp)) in this case.

Clustering for tumor class discovery

Clustering is the process of discovering groups of related examples. In contrast to classification, this process is not guided by knowledge about the proper grouping of some examples. The result of clustering can be described as a set partition of the examples. So the result is a collection of disjoint subsets of the data, whose union is the complete data set again. In tumor class discovery the found subsets are supposed to correspond to discovered tumor subtypes. A different approach is taken in hierarchical clustering. This process leads to a binary tree structure with the examples as leaf nodes. Examples that are connected by internal nodes that are relatively far from the root, can be seen as clusters. A clustering is considered successful if patterns within each group are more similar to each other than to patterns in other groups.

Feature selection for biomarker detection

Feature selection is a procedure which identifies relevant features of the set of patterns. In a supervised setting "relevant" is with respect to the class. A biomarker is an indicator of a disease. As in classification for diagnosis, the

Data type	Laboratory technique
Chromosomal aberrations	Array CGH
Gene expression	Expression array
Protein expression	Seldi-Tof

Table 1.2: Laboratory techniques.

Data type \ Research topic	Diagnosis	Class discovery	Biomarker
Chromosomal aberrations		[27], Chapter 5 and 6	
Gene expression		[27], Chapter 6	
Protein expression	[28], Chapter 7		[28], [26], Chapter 7

Table 1.3: Thesis outline.

class can be chosen to indicate the disease. The relevant features found form a potential biomarker.

1.3 Thesis outline and summary

Since we have analyzed data involving DNA, RNA and proteins it makes sense to use the central dogma of molecular biology as a guide for structuring this thesis. The dogma can be summarized as: *DNA* is transcribed into *RNA* which in turn is translated into *proteins*. Table 1.3 shows which type of data is used for the several cancer research topics in this thesis. The thesis begins with a brief introduction to the molecular biology (chapter 2) and computational methods (chapter 3). The next sections contain the main contributions, depending on the reader's background it may be useful to first read chapter 2 or 3. Finally, a conclusion is presented (chapter 8).

We summarize below the content of the main chapters of the thesis, using the considered data (see Table 1.2) for structuring the presentation.

1.3.1 Chromosomal aberrations

Introduction Chromosomal aberrations that are present in many tumors may indicate the presence of an oncogene or a tumor suppressor gene.

Data Array CGH experiments provide data that should allow us to estimate copynumber changes.

Problem There are many sources of noise in the data. The labeling and hybridization may be uneven. So called cross hybridization causes gains and losses to become less clear. Cross hybridization is the phenomenon that pieces of chromosomal DNA hybridize to random spots. On average this adds an equal amount of material to both channels. If this amount becomes large, the ratio of both channels goes to one. Furthermore, the measured tumor sample may be inhomogeneous. It may contain some normal cells and cells from intermediate stages of the tumor pathogenesis.

Approach To overcome these problems we created a "smoothing" method. It is based on relatively simple assumptions. More sophisticated models could be used if we would have more insight in tumor pathogenesis and the processes involved in the experiments. The approach uses a maximum likelihood criterion and an evolutionary algorithm.

Results The results of the algorithm are comparable to those obtained by an expert. The software to perform the "Smoothing" is available on the website `www.few.vu.nl/~vumarray`. The algorithm has also been incorporated in commercial software.

Chapter 4

Reference [30]

1.3.2 Meta-analysis of array CGH

Introduction Several dual channel array CGH platforms are available today. Comparing experiments from these different platforms may provide valuable insights, however this is not a trivial task.

Data 373 Array CGH primary cancer experiments and 61 cell line experiments from several dual channel platforms: cDNA, BAC and Oligo.

Problem Array CGH data from different platforms cannot be compared directly, since each platform contains different numbers of clones at variable spacing and resolution, and the noise distribution varies across platforms, as well as the amplitude for a given copy number change. Finally, the way each institute collects and processes the spotted elements of the cancer samples may introduce specific noise in the data and influence the dynamic range.

Approach We developed a five-step preprocessing methodology to overcome these problems. Subsequently, a cell line data set was used to optimize the settings for the preprocessing and hierarchical clustering. Next, we applied the procedure to the primary cancer data.

Results We show that currently available array CGH data can be used for a meta hierarchical clustering. Different platforms are used by different

institutes, which results in different types of noise. This is done by, among others, using the smoothing algorithm.

Chapter 5

1.3.3 Clustering gene expression data

Introduction The gene expression profile of a tumor is thought to be one of the most important characteristics of a tumor, since gene expression determines to a large extent the behavior of a cell.

Data We consider five data sets from micro-array gene expression experiments.

Problem A SVM is a powerful technique for classification and regression. It is interesting to apply characteristics of this technique to clustering of gene expression data.

Approach We introduce a heuristic method for non-parametric clustering that uses support vector classifiers for finding support vectors describing portions of clusters and uses a model selection criterion for joining these portions. Clustering is viewed as a two-class classification problem and a soft-margin support vector classifier is used for separating clusters from other points suitably sampled in the data space.

Results The results indicate that the method is a fairly robust clustering method, capable of identifying the "true" structure in the considered data sets.

Chapter 6

Reference [27]

1.3.4 Proteomics

Introduction Proteins determine the behavior of cells. This suggests that proteomics analysis may provide sufficient information to perform a successful diagnosis. Furthermore an understanding of which proteins are important in certain types of cancer is of fundamental relevance.

Data Two proteomic pattern data sets (SELDI-TOF) containing measurements from ovarian and prostate cancer samples.

Problem The problem is to construct a classifier that successfully separates the cancer class of samples from the healthy ones. Moreover, we want to identify those features from the data that are most important for classification.

Approach A linear and a quadratic SVM are applied to the data for distinguishing between cancer and benign status. The prostate dataset is further analyzed by means of an evolutionary algorithm for feature selection that searches for small subsets of features in order to optimize the SVM performance.

Results On the ovarian data set, SVM with all the features exhibits excellent diagnostic performance, while on the prostate dataset it obtains relatively low sensitivity. Results show that the performance of the algorithm on the prostate dataset depends on the data splitting and "run" of the algorithm. Moreover, feature subsets generated vary per run, with a small core of features occurring more often.

Chapter 7

Reference [28]

Chapter 2

Biological background

This chapter provides a minimal background in biology to better understand the remainder of the thesis. A more elaborate introduction can for instance be found in [32].

2.1 Human cells

All organisms consist of cells. Each cell is a system with many different building blocks enclosed in a membrane. Cells can be of different types. For instance there are skin cells, muscle cells and brain cells (neurons).

A human cell has a nucleus, which is separated from the rest of the cell by a membrane. The nucleus contains chromosomes, which are the carrier of the genetic material. Within human cells there are structures, called organelles, e.g., centrioles, lysosomes, golgi complexes, mitochondria among others, which perform particular biological processes. The area of the cell outside the nucleus and the organelles is called the cytoplasm. Membranes are a barrier to the environment, and regulate the flow of food, energy and information in and out of the cell.

Human life begins as a single cell, as a result of fusion of a male and a female sex cell (gametes). The single cell has to grow, divide and differentiate into different cell types to produce tissues and organs. The growth of a cell and its division is called the "cell cycle".

There are three important types of biological macromolecules involved in cells:

1. DNA
2. RNA
3. Proteins

DNA (DeoxyriboNucleic Acid) is the main information carrier molecule in a cell. DNA may be single or double stranded. A single stranded DNA molecule, also called a polynucleotide, is a chain of small molecules, called nucleotides. There are four different nucleotides grouped into two types, purines: adenosine and guanine and pyrimidines: cytosine and thymine. They are usually referred to as bases, since the bases are the only distinguishing element between different nucleotides. They are denoted by their initial letters, A, C, G and T. Any number of nucleotides can be linked together to form a polynucleotide. Specific pairs of nucleotides can form bonds between them. A binds to T, C binds to G. The A-T and G-C pairs are called base-pairs. Two polynucleotides are called complementary, if one can be obtained from the other by exchanging A with T and C with G. Two complementary polynucleotides can stick together. Two complementary polynucleotide chains form a stable structure, which resembles a helix.

RNA is constructed from nucleotides, like DNA. But instead of the pyrimidine thymine (T), it has uracil (U), which is not present in DNA. Messenger RNA is single stranded.

Proteins are the main building blocks and functional molecules of the cell. A protein consists of amino acids joined by peptide bonds. The amino acid chains are folded into 3-dimensional structures. Proteins have varying functions in the cell, some are enzymes, and some have structural or mechanical roles in the cytoskeleton, while others may determine immune response.

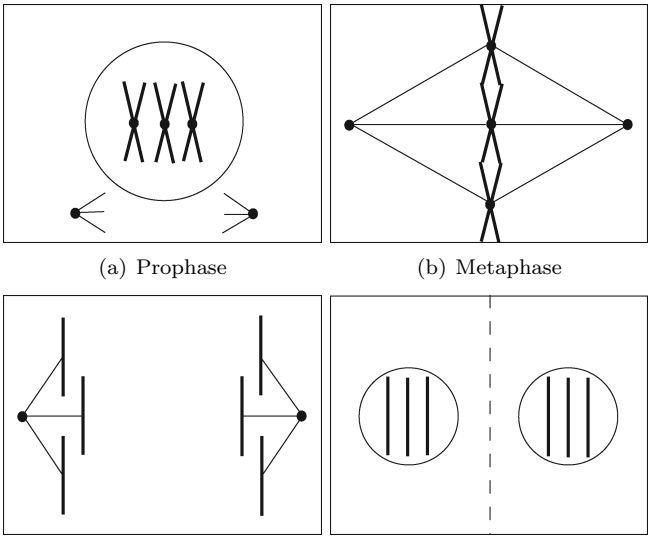
2.1.1 Chromosomes

Chromosomes are strands of DNA. Both ends of the chromosome are called telomeres. A chromosome has two "arms" connected by a centromere. The shortest is called "p", the other "q". Chromosomes are commonly shown in the mitosis state metaphase, when they are duplicated and wound up so that they are visible under an optical microscope. These duplicated chromosomes are called dyads. The duplicates are held together at their centromeres.

All animals have a characteristic number of chromosomes in their cells called the diploid (or 2n) number. These occur as homologous pairs, one member of each pair is acquired from the gamete of one of the two parents of the individual. The complete set of chromosomes in the cells of an organism is its karyotype.

The karyotype of humans contains 22 pairs of so called autosomes. In addition a female has a pair of X chromosomes. A male has one X chromosome and one Y chromosome. The X and Y chromosomes are called the sex chromosomes. The majority of human DNA is identical in all humans.

A gene is a region of chromosomal DNA that controls a hereditary characteristic. It usually corresponds to a sequence used in the production of a specific protein or RNA. A gene carries biological information in a form that is copied from each cell to all its progeny. The locus is a place on a specific chromosome where a gene is found.



(a) Prophase (b) Metaphase

(c) Anaphase (d) Telophase

Figure 2.1: Mitosis.

2.1.2 The cell cycle

The cell cycle is the set of events resulting in cell growth and division. The stages are called G1, S, G2 and M. The G1 stage stands for "GAP 1". In this phase cells increase in size, produce RNA and proteins. The S stage stands for "Synthesis". In this stage DNA replication occurs. The G2 stage stands for "GAP 2". During this phase the cell continues to grow and produce new proteins. The M stage stands for "Mitosis". Mitosis is nuclear division plus cytokinesis, and produces two identical cells in four stages called prophase, metaphase, anaphase, and telophase.

Prophase, see figure 2.1a. The cell contains two so called centrosomes. The two centrosomes move in opposite directions towards the border of the cell. Some fibers (microtubules) grow out of each centrosome. This set of fibers is called the "mitotic spindle".

Metaphase, see figure 2.1b. The membrane around the nucleus disappears. Some protein structure (kinetochore) appears at the centromere of each chromatid (one of two chromosomes after duplication, but before their separation). The microtubules attach to the kinetochores and to the arms of the chromosomes. Both kinetochores are attached to different centrosomes. In the metaphase all the chromatids reach a position midway between the centrosomes.

Anaphase, see figure 2.1c. The two kinetochores at both chromatids separate and each moves to its centrosome dragging its attached chromatid (chro-

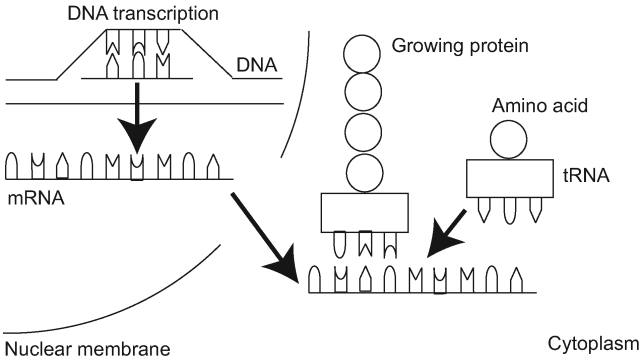


Figure 2.2: Central dogma.

mosome) behind it.

Telophase, see figure 2.1d. A nuclear membrane forms around each of both groups of chromosomes.

Cytokinesis. A ring forms around the cell. As the ring tightens, the cell is divided into two cells.

Obviously, this is a simple summary of the process. However, during this cycle errors can occur, leading to chromosomal aberrations, which in turn can lead to cancer.

2.1.3 The central dogma of molecular biology

DNA contains the complete genetic information that defines the structure and function of a human. Proteins are formed using the genetic code of the DNA. Three different processes are responsible for its conversion from one form to another.

During *transcription* one strand of DNA molecule is copied into a complementary pre mRNA (preliminary messenger RNA). In the process the two-stranded DNA double helix is unwound and information is read from one strand.

Splicing removes some pieces of the pre mRNA, called introns. The remaining sections called exons are joined. An exon is the part of the gene that codes for a protein. The result of splicing is mRNA. The mRNA moves from the nucleus into the cytoplasm.

During *translation* the mRNA sequence is translated into a sequence of amino acids as the protein is formed. The ribosome reads three nucleotides (a codon) at a time from the mRNA and translates them into one amino acid. Transfer or tRNA molecules each carry a specific amino acid to the ribosome and specifically recognizes one codon on the mRNA. The amino acid carried by the tRNA is added to the protein.

These processes, except the splicing, are illustrated in figure 2.2.

2.2 Human cancer

Cancer is a disease in which cell division becomes uncontrolled. Cancer results from molecular events that change the properties of cells. In cancer cells the normal control systems that prevent cell overgrowth and the invasion of other tissues are disabled.

The cells that divide in an uncontrolled way form a tumor. This process can be malignant or benign, meaning life threatening and not dangerous, respectively. A malignant tumor invades surrounding organs and tissues, this process is called metastasis. Tumors from organ cells are called carcinomas. Tumors from the internal or external surface of the body are called adenomas.

The abnormalities in cancer cells usually result from mutations in genes that regulate cell division. Over time more genes become mutated. This is often because the genes that make the proteins that normally repair DNA damage are themselves not functioning normally, because they are also mutated. Consequently, the number of mutations begins to increase in the cell, causing further abnormalities in that cell. Some of these mutated cells die, but other alterations may give the abnormal cell a selective advantage that allows it to multiply much more rapidly than normal cells.

Malfunctioning genes can be classified into three groups. The first group, called proto-oncogenes enhance cell division. The mutated forms of these genes are called oncogenes. The second group, called tumor suppressors prevent cell division. The third group is DNA repair genes, which prevent mutations that lead to cancer.

Aberrations that increase the copy number of oncogenes accelerate growth while those that decrease the copy number of tumor suppressors prevent the normal inhibition of growth. In either case, uncontrolled cell growth occurs.

The conversion of a proto-oncogene to an oncogene may occur by mutation of the proto-oncogene, by rearrangement of genes in the chromosome that moves the proto-oncogene to a new location, or by an increase in the number of copies of the normal proto-oncogene.

Tumor suppressor genes normally inhibit cell growth, preventing tumor formation. Mutations in these genes result in cells that no longer show normal inhibition of cell growth and division.

A third type of gene associated with cancer is the group involved in DNA repair and maintenance of chromosome structure. Environmental factors, such as UV light, and chemicals, can damage DNA. Errors in DNA replication can also lead to mutations. Certain gene products repair damage to chromosomes, thereby minimizing mutations in the cell. When a DNA repair gene is mutated its product is no longer made, preventing DNA repair and allowing further mutations in the cell.

There are several types of chromosomal aberrations that can occur in cancer

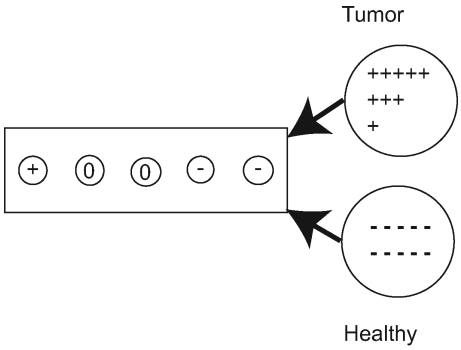


Figure 2.3: Micro-array.

cells. A piece of chromosome can have more than the normal two copies. We say that piece has a gain. A piece of chromosome may also have only one copy or even no copies. This situation we call a loss or double loss. The final type of aberration occurs when a part of a chromosome has moved to a different location. We call this event relocation. It should be noted that not all aberrations cause cancer.

2.3 Micro-arrays

Micro-arrays are glass slides onto which DNA (cDNA or oligonucleotides) are attached, at known locations (probes, spots or targets). An oligonucleotide, or oligo as it is sometimes called, is a short fragment of a single-stranded DNA. Each spot contains material that corresponds to a known gene or location on the genome.

The process is based on hybridization. It uses fluorescently labeled nucleic acid molecules to identify complementary molecules. When two complementary sequences find each other, such as the target DNA and the cDNA in the sample solution, they will stick together or hybridize. Figure 2.3 shows a schematic representation of the process.

There are two types of experiments that can be done. One measures relative chromosomal DNA copy number levels, the other measures relative mRNA or gene expression levels. Some details of the hybridization steps in both methods are given in the next sections. After this hybridization step, the micro-array is placed in a “reader” or “scanner” that consists of some lasers, a special microscope, and a camera. The fluorescent tags are excited by the laser, and the microscope and camera create a digital image of the array.

Image analysis software identifies the spots on the array. Then it calculates intensities of sample DNA, control DNA and background. From these values relative expression or copy number levels can be calculated.

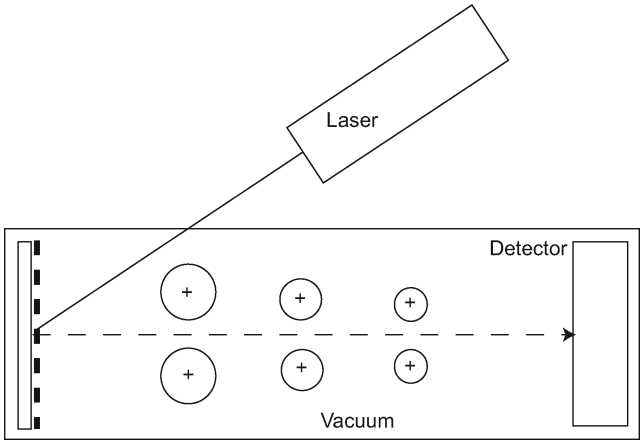


Figure 2.4: Seldi-Tof.

2.3.1 Array CGH

Micro-array Comparative Genomic Hybridization (array CGH) is a technique to look for genomic gains and losses. The hybridization mixture will contain fluorescently labeled genomic DNA obtained from both normal (reference) and cancer tissue. If the number of copies of a particular area has increased, a large amount of sample DNA will hybridize to those spots on the micro-array that represent the area involved in that disease, whereas relatively small amounts of control DNA will hybridize to those same spots. This technique does not notice chromosomal relocations.

2.3.2 Expression arrays

The target DNA is cDNA derived from the mRNA of known genes and the control and sample DNA hybridized to the chip is cDNA derived from the mRNA of normal and diseased tissue, respectively. If a gene is over (under) expressed in a certain disease state, then more (less) sample cDNA, relative to control cDNA, will hybridize to the spot representing that gene.

2.4 Seldi Tof

Proteomics is the study of protein expression and function. Once the proteins are extracted from the sample, a number of techniques are available to separate the different proteins from each other to measure their intensities. Surfaced-Enhanced Laser Desorption/Ionization Time-of-Flight (Seldi-Tof) is one of them.

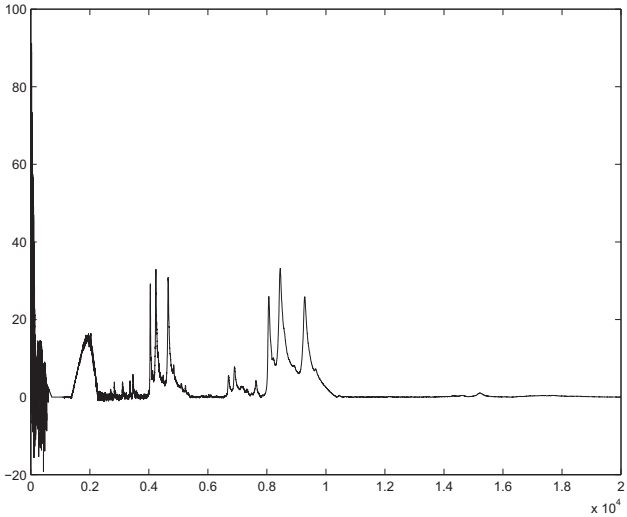


Figure 2.5: A typical protein profile produced by SELDI-TOF MS. The y-axis shows the relative abundance of ionized peptides. The x-axis shows their mass-to-charge (m/z) ratios.

In the Seldi approach a biological sample is spotted onto a solid surface (probe or ProteinChip). Seldi uses selective probe surfaces to capture only a fraction of proteins in a sample. After the sample is put on the probe, the surface is washed to remove all unbound molecules. The probe contains compounds that absorb the energy of light at a particular wavelength. The probe is then inserted into a device with a laser that can fire at the proper wavelength. The energy absorbing molecules transmit some of the energy to the peptides and proteins in the sample. This causes the sample to vaporize. The proteins and peptides are ionized by chemical processes. The ion gas is accelerated into the mass analyzer using electric fields. The amount of kinetic energy for each ion depends on its charge. If two ions have the same charge, their kinetic energy will be the same. Since the velocity given the kinetic energy depends on the weight of the ion, larger ions will move slower. Measuring the time of flight to the detector therefore gives the weight of the ion. Spreads in the amount of initial kinetic energy will cause some inaccuracy. The process is illustrated by figure 2.4.

The Seldi-Tof technology produces a graph of the relative abundance of ionized peptides (y-axis) versus their mass-to-charge (m/z) ratios (x-axis). (Cf. Figure 2.5) The m/z ratios are proportional to the peptide masses, but the technique is not able to identify individual peptides, because different peptides may have the same mass and because of limitations in the m/z resolution. Currently the graph is represented by 15000 measuring points. There is no

obvious relation between neighboring measurement points, apart from the fact that they refer to peptides of similar masses and that the resolution is such that the graph should be considered a smoothed version of the true mass density.

Chapter 3

Machine learning background

3.1 Optimization by evolutionary computation

A more thorough introduction to the subject of this section can be found in [14].

Evolutionary algorithms (EAs) are search methods inspired by natural selection and survival of the fittest in biology. EAs involve a search from a population of candidate solutions. Each iteration of an EA involves a competitive selection that discards poor candidate solutions in a stochastic way. The algorithm constructs variations of the candidate solutions. Two types of variation are recombination and mutation. Recombination refers to the process that recombines candidate solutions randomly with other solutions by exchanging parts. Mutation refers to the process that makes a random change to a single element of a candidate solution. Recombination and mutation are used to generate new solutions that are biased towards regions of the space for which good solutions have already been seen.

Evolutionary computation is traditionally divided into four types of algorithms: genetic algorithms, genetic programming, evolutionary strategies and evolutionary programming. The algorithms are all loosely based on the "survival of the fittest" principle from Darwin's evolution theory.

These algorithms are used for optimization problems where many candidate solutions are possible (large search space), but finding an optimal or even good solution is difficult. If the search space is small an exhaustive search is possible. Other search algorithms are for instance random walk or gradient descent. It seems that evolutionary search performs better than random search and is less vulnerable to local optima than gradient descent.

The main distinctive feature of evolutionary search techniques is their use of a population of solutions, and when recombination is used, the fact that new

```
Evolutionary Algorithm

Generate initial population
Determine fitness of population
Repeat until termination criterion is met
  Select parents
  Apply reproduction operators, producing offspring
  Determine fitness of population
  Select survivors next generation
End repeat
```

Figure 3.1: Outline of an evolutionary algorithm.

solutions are created by two or more old ones.

An important remark for biologists is that the evolutionary computation techniques are based on biological evolution on a high level, but they are not intended to be a realistic model of biological evolution. Aspects of biological evolution can be left out, or extended without biological motivation.

Figure 3.1 gives an outline of a general evolutionary algorithm. The population consists of individuals that represent candidate solutions to the optimization problem. The initial population is filled with some individuals. Then an iteration is started. Each iteration corresponds to a generation in the evolution. The termination criterion typically depends on the maximum individual fitness, or on the amount of change that is observed in the population over the generations. The fitness function indicates the quality of an individual in the population.

The selection of the parent population is based on the fitness of the individuals. Two popular approaches are "roulette wheel" and "tournament selection". In roulette wheel, each individual gets a selection probability proportional to its fitness. In tournament selection a set of k potential parents is chosen at random from the population. The individual with the highest fitness among those individuals is chosen as parent.

A distinction can be made in the reproduction operators. Mutation operators only use one individual to produce offspring. Recombination operators typically use two parent individuals to produce two offspring individuals.

Next to this we need a representation for the solutions. The kind of reproduction operators used is closely related to the chosen representation. The main difference between the four types of algorithms in evolutionary computation lies in these properties.

In genetic algorithms (GAs) the representation of individuals is often a bit string. Operators that are sometimes used are crossover and mutation. Crossover is a type of recombination. It generates two offspring from two parent bitstrings by exchanging parts of the parent bitstrings. Mutation changes one

bit in the bitstring. Mutation is applied after the cross-over.

In genetic programming (GP) the representation is a tree. The nodes are functions, variables or values. Function nodes can have children which represent the arguments of the function. Operators are typically subtree crossover and subtree mutation. Subtree mutation takes from both parents a node (and the trees under them) and exchanges them. Subtree mutation takes one node in the tree and replaces it by another (random) subtree.

In evolutionary strategies (ESs) the representation is a string of real values. A possible recombination operator is intermediate recombination, which generates one offspring from a number of parents by taking the mean of each element of the parent vectors. Gaussian mutation adds a Gaussian generated value to each of the elements. Selection of survivors in ES is divided into two types, named (μ, λ) and $(\mu + \lambda)$. The population is of size μ and λ individuals are selected to be parents in both. In $(\mu + \lambda)$ the λ parents are always chosen to be among the survivors.

In evolutionary programming the representation depends on the problem. The main difference with other evolutionary approaches is that no recombination operators are used, only mutation.

As mentioned earlier, evolutionary computation does not pretend to be a model for biological evolution. An example of this can be found in an application of Lamarckian evolution. The Lamarckian evolution theory describes that traits learned by an individual during its lifetime can be inherited through reproduction by next generations. This theory is not generally accepted by biologists since no biological mechanism seems to be present to achieve this. In evolutionary computation however this can be achieved by applying some specialized search strategies, like local search, to the individuals of the population.

The survivors for the next generation do not need to be selected in a stochastic way. For some problems it may be beneficial always to choose the fittest individual among the survivors. This prevents the accidental removal of a very good candidate solution. This approach is called elitism.

Finally, it should be noted that candidate solutions are often referred to as chromosomes or individuals. In this thesis the word chromosome for candidate solutions may lead to confusion and is therefore used sparsely with this meaning.

3.2 Classification by support vector machines

This section provides a basic introduction to SVMs. More comprehensive introductions can be found in [10, 20].

SVMs can be used for classification. They view the numerical examples as points in a space. The examples belong to two classes. The SVM searches a linear (hyper)plane that separates the points in the space.

If the points of both classes are linearly separable, many potential planes are

possible. The SVM chooses the plane that maximizes the minimum distance of all points from the two classes to the plane.

Extensions to this basic concept include the ability to deal with non-linearly separable data in two ways. The first way is to allow some examples to be on the wrong side of the separating plain. The SVM does include a penalty term that discourages the selection of such planes. The second way is to map the original data into another space by a so called kernel function. This new space is commonly called feature space. The SVM then searches a linear plane in feature space. By proper selection of the kernel function this allows for many types of non-linear separating planes in the original space.

3.2.1 Linearly separable data

The simplest SVM is linear and trained on linearly separable data. Suppose we have a training set of l examples, $\{x_i, y_i\}$ where all $y_i \in \{-1, +1\}$ and $x_i \in \mathbb{R}^d$. The form of an equation that separates the class of points with $y_i = 1$ from the class of points with $y_i = -1$ is $w^T x_i + b = 0$. In this equation w is a weight vector and b is a bias. $w^T x_i + b \geq 0$ if $y_i = +1$ and $w^T x_i + b < 0$ if $y_i = -1$. We call g given by

$$g(x_i) = w^T x_i + b$$

the discriminant or decision function. Points for which the discriminant function is 1 are called support vectors.

Since the data is linearly separable we can find a plane such that $g(x_i) \geq +1$ for $y_i = +1$ and $g(x_i) \leq -1$ for $y_i = -1$, this can be summarized as $y_i g(x_i) \geq 1$. This may require scaling of w and b.

The shortest distance from the hyperplane to the closest positive example is d_+, and for the closest negative example $d\_$. The "margin" of the hyperplane is $d_+ + d\_$. These ideas are illustrated in figure 3.2.

We now would like to know what the distance of x_i is to the hyperplane. Suppose x_p is the normal projection of x_i on the hyperplane and d is the desired distance, see figure 3.3. We can see $x_i = x_p + d\frac{w}{||w||}$. Then $g(x_i) = g(x_p + d\frac{w}{||w||}) = w^T x_p + b + d\frac{||w||^2}{||w||} = 0 + d||w|| = d||w||$. Since support vectors have $|g(x_i)| = 1$, we have $g(x_i) = d||w|| = 1$ for those points. So $d = \frac{1}{||w||}$ and the margin is $\frac{2}{||w||}$.

This means we can maximize the margin between the two classes by minimizing:

$$\frac{1}{2}||w||^2, \text{ such that } y_i(w^T x_i + b) \geq 1, \forall i$$

Support vectors are of particular importance, since the solution would be the same if all non-support vector points were removed.

This constrained optimization problem has a Lagrangian formulation, which in turn satisfies the criteria to allow a "dual" formulation. The goal becomes

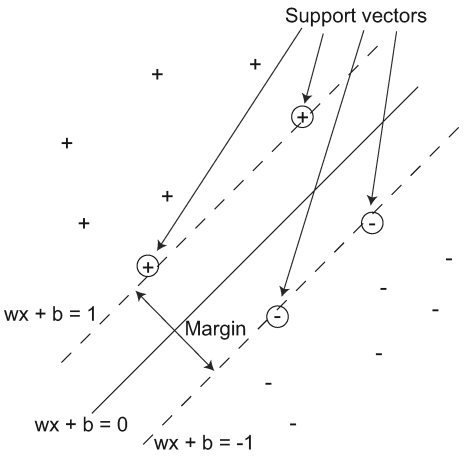


Figure 3.2: Geometric interpretation SVM.

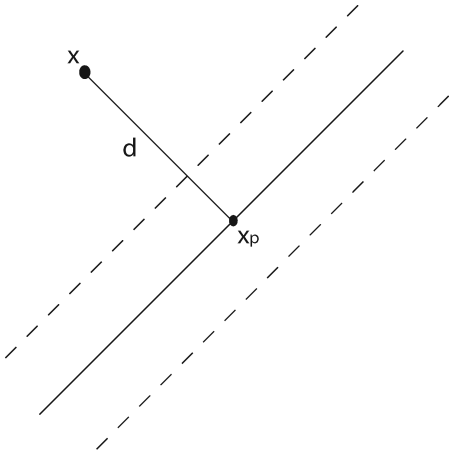


Figure 3.3: Distance to hyperplane.

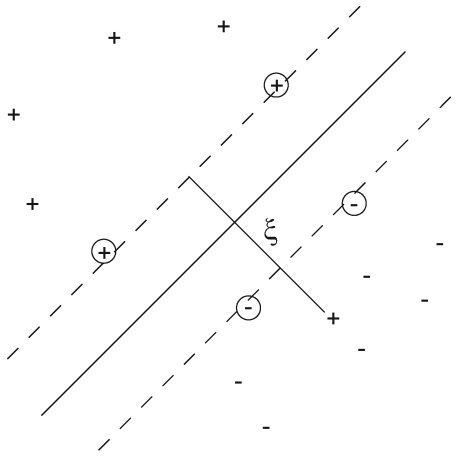


Figure 3.4: Soft margin SVM.

to maximize over the α_i's:

$$\sum_{i=1}^{l} \alpha_i - \frac{1}{2} \sum_{i=1}^{l} \sum_{j=1}^{l} \alpha_i \alpha_j y_i y_j x_i x_j, \text{ such that } \alpha_i \geq 0, \sum_{i=1}^{l} \alpha_i y_i = 0$$

The weights can be retrieved from the Lagrange multipliers α_i and the data set, since $w = \sum_{i=1}^{l} \alpha_i y_i x_i$. A crucial advantage of this formulation is that the data only occurs in the form of dot products.

3.2.2 Non-linearly separable data

The constraints $y_i(w^T x_i + b) \geq 1$ do not allow points to occur beyond the borders of the margin. To soften these constrains, slack variables ξ_i are introduced. The constrains become $y_i(w^T x_i + b) \geq 1 - \xi_i$ and we require all $\xi_i \geq 0$. This situation is illustrated in 3.4. To discourage situations with many misclassifications, the goal becomes to minimize:

$$\frac{1}{2}||w||^2 + C \sum_{i=1}^{l} \xi_i, \text{ such that } y_i(w^T x_i + b) \geq 1 - \xi_i, \xi_i \geq 0 \forall i$$

Again a Lagrangian and its dual can be formulated. We should maximize:

$$\sum_{i=1}^{l} \alpha_i - \frac{1}{2} \sum_{i=1}^{l} \sum_{j=1}^{l} \alpha_i \alpha_j y_i y_j x_i x_j, \text{ such that } 0 \leq \alpha_i \leq C, \sum_{i=1}^{l} \alpha_i y_i = 0$$

Note that the only difference with the linearly separable case is the upper bound on the α_is. In this case a distinction is made between two types of

support vectors. Bounded support vectors have $\alpha_i = C$. Non-bounded support vectors have $0 < \alpha_i < C$.

From optimization theory we can state a number of conditions, known as the Karush-Kuhn-Tucker conditions, that are necessary and sufficient for w, b and the α_is to be a solution to this optimization problem. Some of those conditions are that all $\xi_i(\alpha_i - C) = 0$. This means all non-bounded support vectors will have $\xi_i = 0$. Therefore the non-bounded support vectors are located exactly on the margin boundaries. It should be noted the complete set of Karush-Kuhn-Tucker conditions also allows us to calculate the b in the decision function.

3.2.3 Non-linear SVM

Until now we had a situation where the decision function is a linear function of the data. For some classification problems this approach may not be very suitable. The approach taken in SVMs is to map the data from the input space into another space (feature space), by a function $\phi(x_i)$. If we find a linear hyperplane in the feature space, this may correspond to a non-linear hyperplane in the input space. This approach can of course also be applied to other simpler classifiers such as k-nearest neighbors.

The advantage of SVMs is that the data occurs only as dot products in the optimization problem. This allows the use of kernel functions $K(x, y)$ that are by definition $< \phi(x), \phi(y) >$ for some ϕ. We do not need to specify ϕ explicitly, which allows the mapping to very high dimensional spaces, or even infinite dimensional spaces, for instance by using a Gaussian kernel ($K(x, y) = e^{-\frac{||x-y||^2}{2\sigma^2}}$).

So our goal becomes to maximize:

$$\sum_{i=1}^{l} \alpha_i - \frac{1}{2} \sum_{i=1}^{l} \sum_{j=1}^{l} \alpha_i \alpha_j y_i y_j K(x_i, x_j), \text{ such that } 0 \leq \alpha_i \leq C, \sum_{i=1}^{l} \alpha_i y_i = 0$$

Fortunately we do not require w (which is in the feature space) for the decision function and end up with only dot products again:

$$f(x) = \text{sign}(\sum_{i=1}^{l} \alpha_i y_i K(x_i, x) + b)$$

3.3 Clustering by hierarchical clustering

In hierarchical clustering a binary tree is formed [15]. This is usually done "agglomeratively". Initially each example forms a separate cluster. Then iteratively the two closest clusters are connected to a new parent node which together forms a new cluster, until the tree is complete. This procedure is outlined in figure 3.5.

```
Hierarchical clustering

Put each example in a separate tree (cluster)
Repeat until one tree remaining
  Select ``closest'' pair of trees
  Connect roots of those trees to new root node
End repeat
```

Figure 3.5: Outline hierarchical clustering.

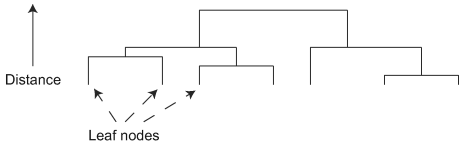


Figure 3.6: Typical output hierarchical clustering.

The distance between clusters is based on the distance between individual examples. Some methods look at the set of distances between each pair of examples, where each example is from a different cluster. "Single linkage" takes the minimum of the set as cluster distance. "Complete linkage" takes the maximum. "Average linkage" takes the average distance over pairs in the union of both clusters.

The distances as obtained by the different linkage methods can be described more formally. Consider X and Y to be sets of examples that occur in separate trees. The single linkage distance between X and Y is $\min\{d(x,y)|x \in X \text{ and } y \in Y\}$. Their complete linkage distance is $\max\{d(x,y)|x \in X \text{ and } y \in Y\}$. Finally the average linkage distance is $\text{mean}\{d(x,y)|x \in X \text{ and } y \in Y\}$.

To measure the distance between the pairs of examples, any metric will do. The measure chosen depends on the application. Popular choices include the Euclidian distance, and $1 - r$, where r is the Pearson correlation coefficient of the two examples.

Figure 3.6 shows a typical tree or dendrogram produced by hierarchical

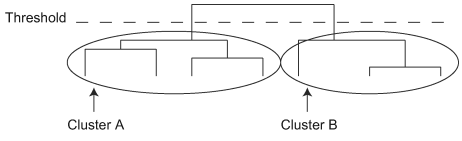


Figure 3.7: Forming clusters from the dendrogram.

```
Wrapper Feature Selection Algorithm

Repeat until termination criterion is met
  Select feature set
  Test feature set using learning method
End repeat
```

Figure 3.8: Outline wrapper feature selection algorithm.

clustering. The position where two clusters are joined depends on the distance between the clusters.

Figure 3.7 illustrates how the dendrogram can be used to obtain a true clustering. A threshold is chosen thus dividing the large tree in several smaller trees. The examples in each tree form a separate cluster.

3.4 Feature selection by wrapping

In general feature selection selects the relevant features from the data to describe the target concept. So it is a reduction of the dimensionality of the data.

There are many approaches in machine learning and statistics that can be thought of to perform some kind of feature selection. For instance, decision tree learning has some implicit feature selection. Unfortunately decision trees are not very well capable of dealing with continuous attributes. Principle components analysis (PCA, [20]) constructs features as linear combinations of variables in the data in such a way that only a small number of those features are required to describe the target concept. Unfortunately, in the data sets used in this thesis the variables correspond to well defined biological concepts like, pieces of chromosomal DNA, genes and proteins. The variables have a clear biological interpretation. Linear combinations of those variables are much harder to interpret. PCA does not clearly include or exclude features, it assigns a weight to them.

For the tasks addressed in this thesis neither of the previous examples are very suitable. In this thesis the problems have continuous variables and the results should have a clear biological interpretation. The examples do indicate there is a great variety of possible approaches to feature selection.

An important distinction can be made between so called “wrapper” and “filter” methods. The wrapper approach of feature selection uses a search algorithm to identify candidate subsets and the actual learning method is used as a “black box” to evaluate the quality of the subset. The filter method does not use the actual learning method to select the features. In this thesis emphasis is on wrapper methods. The most general outline of a wrapper algorithm is

given in figure 3.8.

When developing a wrapper method an important question is how to move through the search space. Possible approaches are to start with one feature and iteratively add one new feature. This is called "forward selection". Another way is to start with all features and iteratively remove one. This is called "backward elimination".

Another question is how to structure the search. An exhaustive search may not be possible. There are many possibilities, like a greedy approach that never reconsiders a choice when using one of the above mentioned search space traversal strategies. Another method is called "stepwise" selection or elimination. At each decision point you consider to remove or add a feature. Best first search may also be possible.

Furthermore a stopping criterion for the search is required. Within wrapper approaches the algorithm can stop if none of the feature sets that are considered for the next iteration is an improvement.

Chapter 4

Noise reduction in array CGH

4.1 Introduction

Array CGH is an approach for genome-wide measurement of aberrations in chromosomal copy numbers. As mentioned in chapter 2, normal human cells contain two copies of each of the 22 non-sex chromosomes (autosomes). In tumor cells one or both copies of parts of chromosomes may be deleted or duplicated. Chromosomal copynumbers are defined to be 2 for normal cells, 1 or 0 for single and double deletions and 3 and higher for single copy gains and higher level amplifications.

The goal of the array CGH technique is the detection of DNA sequence copy number changes and determination of the associated breakpoints along the chromosomes. In other words, the purpose of array CGH is to construct a graph of the copynumbers for a selection of clones (small pieces of DNA) as a function of position of the clone on the genome.

DNA copynumber aberrations are used in cancer research, for instance, by searching for novel genes implicated in cancer by analyzing those genes located in regions with abnormal copy numbers. It is therefore of fundamental relevance to identify as precisely as possible chromosomal regions with abnormal copy numbers.

Because chromosomal DNA copy numbers for technical reasons cannot be measured directly at high resolution, DNA from test cells is directly compared to DNA from normal cells, using several thousands of small DNA fragments, with known identity and genomic position (frequently referred to as clones or BACs), as probes. Every single experiment yields tumor to normal ratios for each clone on the array, and thus for each chromosomal location (see [47]).

4.1.1 Array CGH experiments

In this section we will formalize a simple model of an array CGH experiment. This should allow us to better understand some of the difficulties involved in this type of experiments.

The outline of the array CGH procedure is as follows. Because copy numbers cannot be measured directly, tumor cells are compared to normal cells. A large number (thousands) of clones of different genomic positions are printed on spots on a glass slide (micro-array), which is next treated with a mixture of DNA originating from tumor and normal cells, both labeled and divided into fragments. Before applying the DNA mixture to the micro-array the two types of DNA are labeled red (Cy5) and green (Cy3), respectively. The labeled fragments hybridize ("stick") to a spot on the array with a matching DNA sequence. The measured red/green ratio for each of the spots on the array is roughly proportional to the quotient of copy numbers for tumor and normal tissue. This experiment is repeated for a number of tumors. Below a more elaborate treatment of the procedure is given.

A sample of cells taken from a tumor will generally consist of multiple cell types, which may differ in their chromosomal copy numbers. In particular, the sample usually consists of tumor cells and admixed normal cells. In some cases the sample may also contain tumor cells that are intermediate in the development of the tumor. Suppose the tumor sample consists of T different cell types. Each of the T cell types has a certain copy number for each of the n chromosomal locations measured. We call the copy number x_{ij}, where i is the cell type and j the genomic position. Suppose that the chromosomal material of N_i cells of type i is used in the experiment.

Suppose that in the experiment M reference cells are used. Obviously, the reference cells are normal cells that have copy number 2 for all genomic positions.

The micro-array contains a number of spots (can be thousands) that each corresponds to a specific genomic position. Each spot contains a number of pieces of identical cDNA, called clones. Suppose that each spot contains H_j clones. This number depends on the particular spot, because there may be a slight variation in the weight of the material put on the spot. Below we will provide more details of the experiment.

The tumor and reference DNA is cut into small fragments and labeled. Suppose all fragments have about an equal length of l_f. Furthermore suppose all clones of all spots have about an equal length l_c. That gives $\sum_{i=1}^{T} \frac{N_i x_{ij} l_c}{l_f}$ fragments of tumor DNA that could correctly fit a clone of spot j. There are $\frac{2Ml_c}{l_f}$ fragments of reference material that can correctly fit to spot j. Obviously some fragments cannot fit correctly to any clone, because the genomic position of that fragment is not measured.

The tumor fragments are labeled red (with the Cy3 molecule). The reference fragments are labeled green (with the Cy5 molecule). This labeling may not be equally successful for the Cy3 and Cy5 molecules. The Cy3 and Cy5 molecules

will attach to the Cs in the fragments. Suppose that r is the fraction of Cs that gets labeled with Cy3 and g the fraction that gets labeled with Cy5.

After the labeling the fragments are put on the micro-array. It is assumed an equal weight of fragments of tumor and reference samples is used on the spots. So the actual number of cells used is not known. This introduces a problem. For example, if all chromosomes of all tumor cells have copy number four, we cannot distinguish them from those of normal cells, because simply half the amount of tumor cells would be used in the experiment giving an equal weight.

Not all fragments that could correctly hybridize to a specific clone type actually do. Some of them will be washed of at the end of the experiment. Suppose a fraction f_s does correctly hybridize. This fraction is thus assumed to be independent of the clone type or the difference in Cy3 or Cy5 labeling. This gives $rH_jf_s\sum_{i=1}^{T}\frac{N_ix_{ij}l_c}{l_f}$ tumor fragments that will correctly hybridize. There are $gH_jf_s\frac{2Ml_c}{l_f}$ reference fragments that correctly hybridize.

It should also be noted that the Cs may actually not be uniformly distributed over the genome. Suppose there are c_j Cs in location j. This would give $rc_jH_jf_s\sum_{i=1}^{T}\frac{N_ix_{ij}l_c}{l_f}$ labeled C's of tumor fragments hybridized to spot j. There are $gc_jH_jf_s\frac{2Ml_c}{l_f}$ such Cs in the reference.

Furthermore there are some fragments that will "cross hybridize", that is hybridize to a random genomic position. This is caused by the fact that there are some sequences on the genome that are somewhat similar. Suppose for all spots we find a number of such fragments with in total w_j labeled Cs for both the tumor and reference samples. This number may depend on the genomic location, because not all locations may have an equal amount of similar enough other regions on the genome to cause cross hybridization.

On the spots we can measure the red/green intensity ratio, which would be:

$$\frac{rc_jH_jf_s\sum_{i=1}^{T}\frac{N_ix_{ij}l_c}{l_f}+w_j}{gc_jH_jf_s\frac{2Ml_c}{l_f}+w_j}$$

If we assume almost no cross hybridization, this reduces to:

$$\frac{r\sum_{i=1}^{T}N_ix_{ij}}{2gM}$$

If we assume there is one cell type that is by far most common we can reduce this even further to $\frac{rnx_j}{2gM}$.

One of the reasons why it is good to perform these experiments relative to a normal reference is that the value for a particular spot becomes less sensitive to the amount of material on the spot (H_j). Another reason may be that it makes the values less sensitive to the distribution of Cs over the genome (c_j). Finally, the effect of cross hybridization is reduced to a linear scaling problem. It is important that genomic locations measured within one experiment are directly comparable to each other.

From the previous elaboration it seems the array CGH ratios should be proportional to the actual copy numbers. The CGH scale between different experiments may vary due to the n, M, H, r, g, c_j, f_s and w, based on this model. Estimating these variables is complicated since in many cases chromosomal positions are not repeated on different spots. It should be noted this model is very limited and is intended to illustrate some of the peculiarities of array CGH data. There may be many more experimental factors involved.

A common way to scale is to divide by the median measured ratio. In the sections about smoothing and clustering of experiments we will assume the ratios are normalized in this way (sometimes also log transformed).

4.2 Smoothing

"Smoothing" refers to the process that adjusts the observed array-CGH values such that they represent the copy number of the most common tumor cells. That is, the algorithm tries to set the values to the means of the "clouds" of points that are visible from the array CGH plots. Since the copy number of a clone is always quite small (normally 2, varying from 0 to about 10), we would like to set means of "clouds" that are close to the same value, because they represent the same copy number. Next we also want the number of value changes ("breakpoints") to be small.

4.2.1 Algorithm

The problem can be formalized as model fitting to search for most-likely-fit model given the data. A model describes a number of breakpoints, a position for each, and parameters of the distribution of copy number for each. Then one has to estimate the real parameters of the model from the observed array-CGH values.

We assume that the data are generated by a Gaussian process and use the maximum likelihood criterion for measuring the goodness of a partition, adjusted with a penalization term for taking into account model complexity. We introduce a local search procedure that searches for a most probable partition of the data using N breakpoints, for a given N. The procedure is incorporated into a genetic algorithm that evolves a population of partitions with possibly different number of breakpoints that may vary during execution. We design two algorithms based on this approach. The first one is a genetic local search algorithm that iteratively selects two 'good' chromosomes, generates two offspring using uniform crossover, applies mutation and the local search procedure to the offspring and replaces the worst chromosomes of the population with them. The second algorithm generates only one offspring, applies local search and tries to further optimize the offspring with an ad-hoc procedure.

We analyze the performance of these algorithms on array-CGH measurements for 9 gastric cancer tumors. For each chromosome of a tumor, we com-

```
LS
{
 Take random set of breakpoints among chromosomal positions
 While fitter set of breakpoints can be found
 {
  For all breakpoints (in random order)
  {
   Move breakpoint one chromosomal position to left
     (or right by flipping coin)
   If set of breakpoints becomes fitter
   {
    Keep new set of breakpoints
    Start while loop again
   }

   Move breakpoint one chromosomal position to right
   If set of breakpoints becomes fitter
   {
    Keep new set of breakpoints
    Start while loop again
   }
  }
 }
}
```

Figure 4.1: Outline LS algorithm.

```
SA
{
 Take random set of breakpoints among chromosomal positions

 While fitter set of breakpoints can be found
 {
  For all breakpoints (in random order)
  {
   Move breakpoint one chromosomal position to left
     (or right by flipping coin)
   If set of breakpoints becomes fitter or random accept
   {
    Keep new set of breakpoints
    Start while loop again
   }

   Move breakpoint one chromosomal position to right
   If set of breakpoints becomes fitter or random accept
   {
    Keep new set of breakpoints
    Start while loop again
   }
  }
 }
}
```

Figure 4.2: Outline SA algorithm.

pare the smoothing of the two GAs, the multi-start LS and the multi-start SA algorithm. The LS algorithm is outlined in figure 4.1. The SA algorithm is outlined in figure 4.2. Both algorithms are explained in more detail later. The best algorithm we compare with the expert.

4.2.2 Breakpoint detection

In the used CGH experiments copy numbers are measured for approximately 2200 clones spread along the genome. We apply our algorithm to each of the 23 chromosomes separately. Denote by $x_1, ..., x_n$ the measured CGH values for a given chromosome. The main goal is to cluster these values in a small number of clusters $(x_1, ..., x_{y_1}), (x_{y_1+1}, ..., x_{y_2}), ..., (x_{y_N+1}, ..., x_n)$ such that the copy numbers of the clones in each cluster are identical. We refer to the indices $y_0 = 0 < y_1 < ... < y_N < n = y_{N+1}$ as breakpoints.

Our algorithm is motivated by the working hypothesis that the measured value x_j is equal to the relative copy number of clone j plus random noise that is independent across clones. Thus our model stipulates that for $y_{i-1} < j \leq y_i$ the observed CGH value x_j can be considered as drawn from a normal distribution with mean μ_i and variance σ_i^2 particular to the ith cluster. This leads to the likelihood function

$$\prod_{i=1}^{y_1} \frac{1}{\sigma_1\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{x_i-\mu_1}{\sigma_1}\right)^2} ... \prod_{i=y_N+1}^{n} \frac{1}{\sigma_{N+1}\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{x_i-\mu_{N+1}}{\sigma_{N+1}}\right)^2}$$

The maximum likelihood estimators are the parameter values for which this expression is maximal. Given breakpoints $y_0 = 0 < y_1 < ... < y_N < n = y_{N+1}$ the maximization relative to the μ_i and σ_i^2 is equivalent to performing maximum likelihood estimation on each of the samples $x_{y_{i-1}+1}, ..., x_{y_i}$ separately, which leads to the usual estimates

$$\hat{\mu}_i = \frac{1}{y_i - y_{i-1}} \sum_{j=y_{i-1}+1}^{y_i} x_j$$

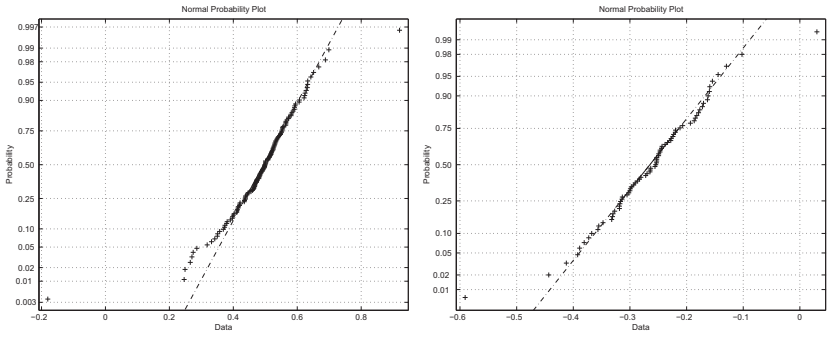
and

$$\hat{\sigma}_i^2 = \frac{1}{y_i - y_{i-1}} \sum_{j=y_{i-1}+1}^{y_i} (x_j - \hat{\mu}_i)^2$$

Reinserting these values into the likelihood we are, after some simplification, left with

$$\frac{1}{\hat{\sigma}_1^{y_1}\sqrt{2\pi}^{y_1}} e^{-\frac{1}{2}} ... \frac{1}{\hat{\sigma}_{N+1}^{n-y_N}\sqrt{2\pi}^{n-y_N}} e^{-\frac{1}{2}}$$

The next step is to find suitable breakpoints by maximizing this relative to $y_1, ..., y_N$. Equivalently, we minimize minus the logarithm, which up to an additive constant is equal to



(a) Chrom 8, tumor tissue, a gain (b) Chrom 14, tumor tissue, a loss

Figure 4.3: Normal probability plots [52]

$$\sum_{i=1}^{N+1} (y_i - y_{i-1}) \log \hat{\sigma}_i$$

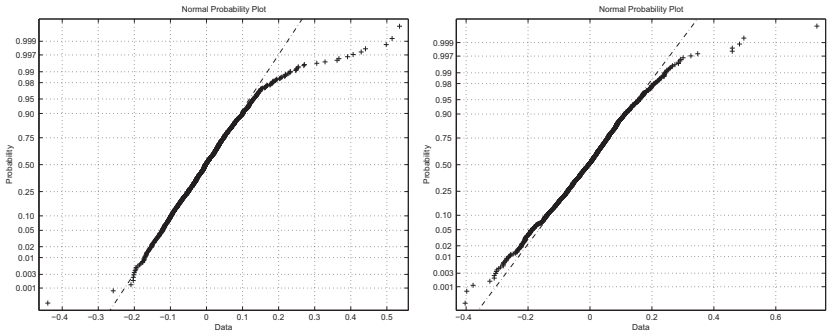
Note here that the $\hat{\sigma}_i$ in this expression also depend on the choice of $y_1, ..., y_N$. However, it is obvious that the highest value of the likelihood is obtained by choosing the highest possible number of breakpoints, as this gives more flexibility in choosing the parameters μ_i and σ_i. The last minimization step is therefore not well defined. We remedy this by adding a penalty to the criterion, in order to discourage a large number of breakpoints. A simple penalty of the form λN, for λ a suitable constant, performed well in our experiments. This leads to the following function to be minimized.

$$f(y_1, ..., y_N) = \sum_{i=1}^{N+1} (y_{i+1} - y_i) \log \hat{\sigma}_i + \lambda N \tag{4.1}$$

If we consider there to be $3N$ parameters ($2N$ continuous parameters and N breakpoints), then the choices $\lambda = \frac{3}{2} \log n$ and $\lambda = 3$ correspond to Bayesian information criterion [57] and Akaike information criterion [1], respectively. In our experiments the choice $\lambda = 10$ was appropriate.

The assumptions of normality and independence may be slightly violated, as illustrated by the normal probability plots [52] in figure 4.3 (data are normally distributed when they lay near the dashed line). This holds also for normal tissues, as shown by the plot in figure 4.4.

Nevertheless, in our experiments the resulting criterion gives adequate results.



(a) Chrms 1 to 22, tumor tissue, all normal

(b) Chrms 1 to 22, normal tissue

Figure 4.4: Normal probability plots [52]

4.2.3 The genetic / local search algorithms

The local search algorithm takes as input the CGH data $l = x_1, ..., x_n$ of one chromosome, a number N of randomly generated indices $y_1, ..., y_N \in [1, n]$ indicating potential (locations of) breakpoints, and updates repeatedly the breakpoints (locations) in order to minimize the function f given in (4.1), where the first term is the negative log-likelihood of the data and the second one is a penalization with parameter λ which penalizes partitions containing many breakpoints. The algorithm uses f as scoring function. At every iteration an update rule is applied to each breakpoint, selected randomly. The update rule chooses randomly a direction (left or right) and moves the breakpoint location of one position in that direction only if the move improves the scoring value (that is if f decreases), otherwise it moves the breakpoint of one place in the opposite direction if this yields an improvement. The iterative process terminates when the application of the update rule to each breakpoint does not improve the scoring. We call this algorithm LS. We use LS in a multi-start local search algorithm, and as local optimizer in the two heuristic algorithms described in the sequel.

Genetic local search algorithms, also called memetic algorithms [38], use local search for optimizing the population after or during the application of the genetic operators. So at each iteration of the evolutionary process the population consists of a set of local optima. We introduce the two memetic algorithms illustrated below for identifying breakpoints in array-CGH data of a chromosome, called GLS (figure 4.5) and GLSo (figure 4.6), respectively.

In order to avoid confusion, in the sequel we say "individual" instead of the standard genetic algorithms term "chromosome" for indicating an element of the population.

Our genetic algorithms use a representation where an individual is a bit

```
GLS
{
 generate initial population
 while (termination criterion not satisfied)
 {
  select two parents from population using roulette wheel
  generate offspring using uniform crossover
  apply mutation to each offspring
  apply LS to each offspring
  replace two worst individuals of population with offspring
 }
}
```

Figure 4.5: Outline GLS algorithm.

```
GLSo
{
 generate initial population
 while (termination criterion not satisfied)
 {
  select two parents from population using roulette wheel
  generate offspring using OR crossover
  apply LS to offspring
  apply JOIN to offspring
  replace worst individual of population with offspring
 }
}
```

Figure 4.6: Outline GLSo algorithm.

string denoting chromosome locations with a 1 in each location containing a breakpoint and a 0 elsewhere. To maximize the fitness function, the score function (4.1) has to be minimized, i.e. best fitness equals minimal score. The initial population is constructed as follows. For each N in a fixed range, a number k of elements is generated, where an element is a bit string with N 1's randomly placed. The local search LS is applied to each individual.

GLS uses (blind) uniform crossover, while mutation randomly decides whether to add or remove a breakpoint and then applies the chosen operation (that is flipping the value of the selected individual location). The "remove" operation consists of removing the breakpoint that yields the best fitness. Note that this operation is applied even if it does not improve the fitness of the individual. The "add" operation selects the segment (a region between two consecutive ones) with relative chromosomal array-CGH region (set of clones values) having the highest standard deviation, and places a breakpoint in the middle of that region.

The termination criterion is satisfied when either a maximum number of iterations is reached or when the score of the best individual does not decrease (or, equivalently, its fitness does not increase anymore) and there is no pair of corresponding clones in the population having a difference in smoothed value of more than 0.01. The smoothed value of a clone is the mean value of the (chromosomal array-CGH region corresponding to the) segment containing that clone.

GLSo generates one offspring per iteration by selecting two individuals and constructing one offspring by taking the union of their breakpoints (by performing a bitwise OR of the two individuals). Then the offspring is optimized using LS and further optimized by removing breakpoints using the JOIN procedure. The JOIN procedure repeatedly selects the breakpoint whose removal yields the biggest improvement (decrease) of the fitness function, and continues until the fitness does not decrease anymore.

4.2.4 Experimental results

Genomic DNA was isolated from snap-frozen tumor samples taken from gastrectomy specimens. The samples were obtained from the archives of the department of Pathology of the VU University Medical Center. Array-CGH experiments were performed according to [60] and ratio measurements according to [24]. The scanning array comprised DNA from 2275 BAC and P1 clones spotted in triplicate, evenly spread across the whole genome at an average resolution of 1.4 Mb. Chromosome X-clones were discarded from further analysis since all tumor samples were hybridized to male reference DNA, leaving 2214 clones per array to be evaluated. Each clone contains at least one STS ("sequence-tagged site", a unique DNA sequence of a few hundred base pairs long) for linkage to the sequence of the human genome. These data are analyzed in [68].

The 9 tumors used to test our method are all gastric tumors. A manual

Algorithm	Median	Mean
mLS	-192.99	-218.77
mSA	-193.29	-220.07
GLSo	-194.47	-220.83
GLS	-196.00	-223.08

Table 4.1: Mean and median fitness values for all algorithms. (Smaller is better)

smoothing for these tumors, carried out by the expert B. Ylstra, is used to assess the performance of the algorithms and the maximum likelihood function as approximation for the expert. We run our algorithms on each chromosome of these 9 tumors, for a total of 207 chromosomes containing an average of about 100 clones.

The following GA parameter setting is chosen. The initialization generates 40 individuals containing N breakpoints, with N that varies from 1 to 10. An individual with 0 breakpoints is also added, thus giving a total of 401 individuals. The maximum number of local searches allowed is 100000. The crossover rate equals 1. A single bit mutation is always applied.

The multi-start LS performs 100000 plus 1 local searches, where the number N of breakpoints varies from 1 to 20, with an equal number of runs assigned to each value of N. Also a run with 0 breakpoints is done. The final result is the solution with best score over the runs. Note that the number of local searches is kept equal, to make a fair comparison.

We compare the performance of GLSo, GLS, multi-start LS, and a multi-start variant of LS based on simulated annealing (SA). The annealing schedule of SA is as follows. The starting temperature is chosen to be 100000 (10^5). Furthermore, it is chosen that after 10000 (10^4) changes of breakpoint location the temperature should cool down to 0.00001 (10^{-5}). This implies that after each change the actual temperature is divided by $10^{10^{-3}}$, since $\frac{10^5}{x^{10^4}} = 10^{-5} \Rightarrow x^{10^4} = \frac{10^5}{10^{-5}} \Rightarrow x^{10^4} = 10^{10} \Rightarrow x = (10^{10})^{10^{-4}} \Rightarrow x = 10^{10^{-3}}$. After 10000 changes we make the algorithm behave exactly like LS. The other settings are similar to the multi-start LS, except that it only performs 2001 runs to make the comparison fairer in terms of computation time. The SA algorithm is outlined in figure 4.2, where "random accept" means that a set of breakpoints is accepted with a probability inversely proportional to the temperature.

We compare the performance of the four algorithms in minimizing the function (4.1) by the median and mean fitness values obtained for the $9 \times 23 = 207$ chromosomes in our gastric tumors. In table 4.1 mLS and mSA denote multi-start LS and SA, respectively. As shown in table 4.1 method GLS performs best according to this criterion followed by the second genetic algorithm GLSo.

At closer inspection the nature of the differences in performance of the four algorithms vary considerably over the 207 chromosomes. For 22 chromosomes all four algorithms yield an identical smoothing, and for as many as 76 chro-

Algorithm	Median	Mean	Trimmed Mean	# Zeros	P-value
mLS-mSA	0.00	1.30	0.52	67	0
mLS-GLSo	0.00	2.06	0.22	50	0.63
mLS-GLS	0.70	4.31	1.84	81	0
mSA-GLSo	0.00	0.76	-0.08	31	0.47
mSA-GLS	0.79	3.01	1.66	62	0
GLSo-GLS	0.24	2.25	1.50	48	0

Table 4.2: Differences fitness values for all algorithms

mosomes the smoothings produced by at least three of the four algorithms are identical. For this reason we also investigated the differences in fitness for the 207 chromosomes for each pair of algorithms. Medians, means, 20% trimmed means, and the number of chromosomes with identical smoothing are given in table 4.2, together with the p-values of the sign test. The latter test (cf. [52]) indicates if the observed differences (in obtained fitness values) between the pairs of algorithms are statistically significant, under the assumption that the 207 chromosomes can be considered a random sample of chromosomes.

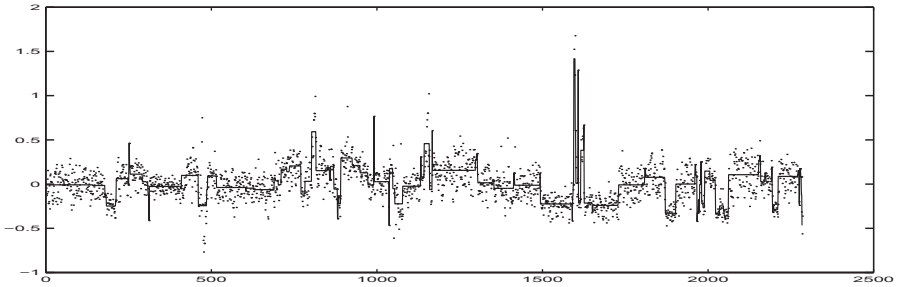
From these numbers we conclude again that GLS is best, followed by GLSo, mSA and mLS. Furthermore, the superior performance of GLS is statistically significant, whereas the observed differences between the other methods may not be replicable on data of additional tumors.

To illustrate the results of the four algorithms we show pictures (figure 4.7) of the smoothing/breakpoints found by each of algorithms on one of the tumors in which various types of chromosomal aberrations (gain, loss and amplification) occur.

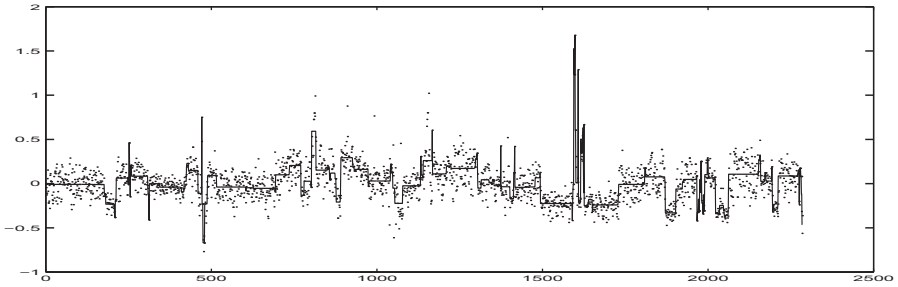
In order to assess the convergence behavior of the genetic algorithms we give plots (figure 4.8) of a typical histogram of the distribution of breakpoints within the solutions of the pool after the stopping criterion is satisfied. The plots indicate that the evolutionary process ends with individuals with breakpoints in nearby locations. Observe that the stopping criterion is such that shifting a breakpoint within an area that has no clear breakpoint stops the iterations. In such a case there is "no clear breakpoint", meaning that the means of the two corresponding segments in all individuals are close. This may cause the algorithm to stop after a few iterations even if the individuals have breakpoints in different locations.

Next, we compare the robustness of the genetic algorithms, that is the sensitivity of the outcome to the initialization and other random operators used. We plot a typical histogram (figure 4.9) of the location of breakpoints of the best individual of the final population over 100 runs of the genetic algorithms on chromosome 1.

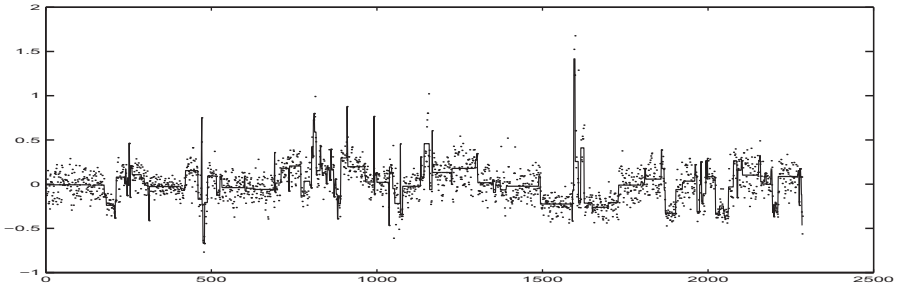
Finally, we compare the smoothings and breakpoints obtained by GLS with those manually produced by the expert. The manual smoothings have been



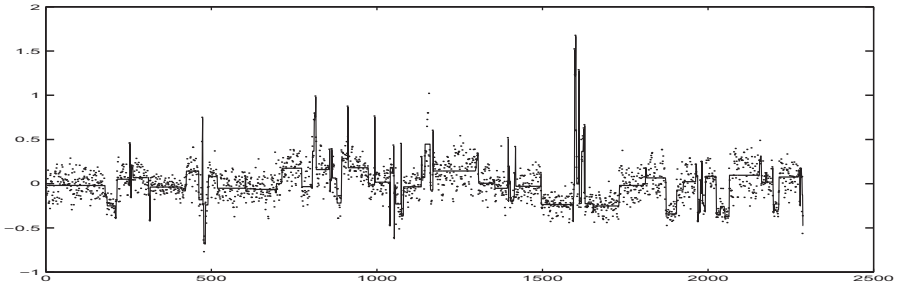
(a) Tumor 2008c, dots are raw data, line is result of multi-start LS



(b) Tumor 2008c, dots are raw data, line is result of multi-start SA

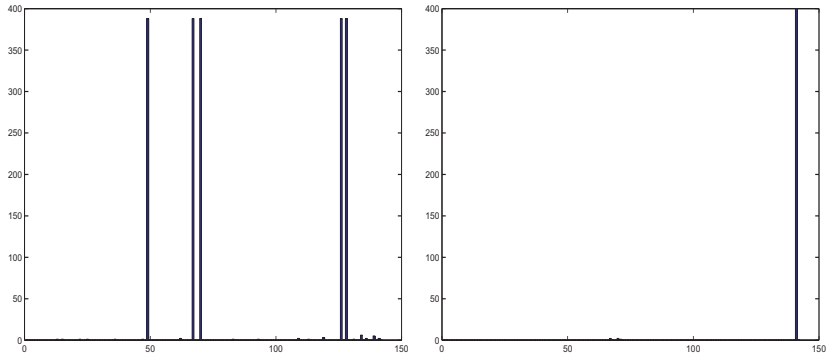


(c) Tumor 2008c, dots are raw data, line is result of GLSo



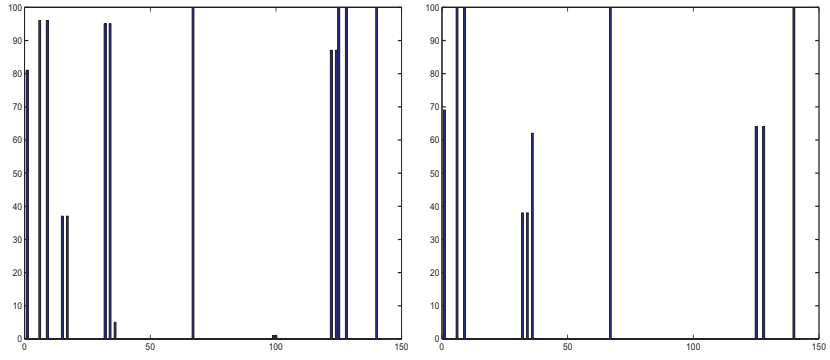
(d) Tumor 2008c, dots are raw data, line is result of GLS

Figure 4.7: Illustration results all algorithms



(a) GLSo, tumor 2730, chrm 1 (b) GLS, tumor 2730, chrm 1

Figure 4.8: Distribution of breakpoints in pool after stopping criterion



(a) GLSo, tumor 2730, chrm 1 (b) GLS, tumor 2730, chrm 1

Figure 4.9: Distribution of breakpoints for best individual of final population over 100 runs

built under the assumption that there is a small number of different smoothing levels, reflecting the observation that few copy number values are present in chromosomes. In order to incorporate this constraint in our method, we perform a post processing step that joins close smoothing levels. To this aim the k-means algorithm is applied to the set of CGH values generated by running GLS over all the chromosomes of a tumor, and then the resulting smoothing levels that are closer than a fixed threshold are joined. Over all tumors the average difference between the values of the clones is 0.0513, indicating that GLS followed by post processing (denoted below by GLS-pp) is a satisfactory approximation of the manual smoothing. GLS seems more sensitive to outliers. This can be explained by the fact that the expert sometimes knows an outlier is meaningless and so ignores it. Figure 4.10 shows an example comparison between our algorithm and an expert smoothing.

4.2.5 Discussion

The results of the experiments indicate that GLS performs better in minimizing function (4.1) than the other algorithms.

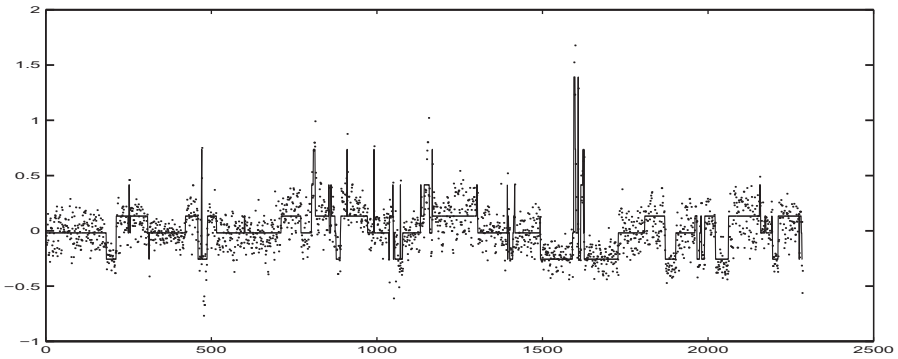
Both GAs converge within the maximum number of iterations in case the data contains clear breakpoints. The stopping criterion prevents the algorithm from searching for optimal locations of breakpoints that are not clear.

GLS-pp finds smoothings that are very similar to the manual smoothings. It should be noted that an expert produces smoothings based on more information than just the CGH values. An expert also keeps in mind information like misplacement of clones on the genome and recurring aberrations of clones due to known experimental artifacts. From the normality plots shown it seems that the final expert smoothings are reasonably well normally distributed. Lacking data combining CGH values with known copy numbers in cell types and frequencies of cell types in samples, we were not able to test the suitability of our model to remove noise from the experiment and some cells of types that occur in small numbers. This remains an open problem for future research.

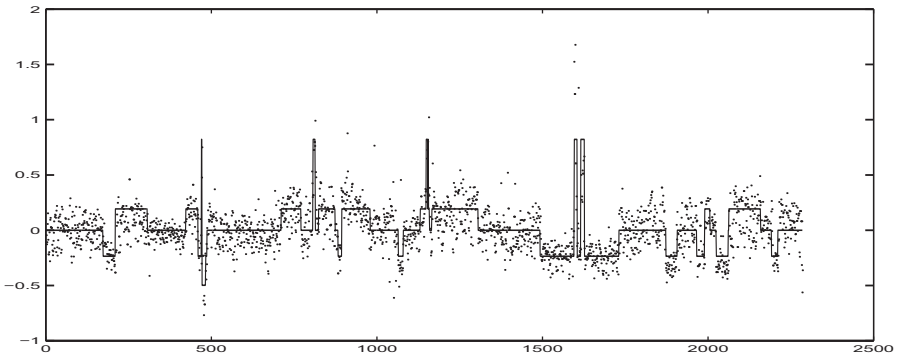
4.3 Software

The ratios found in array CGH experiments have some noise generated by polymorphic sites (sequence variation between individuals), some experimental noise as well as compression of the ratios due to aneuploidy and admixed non-tumor cells. This noise renders the identification of breakpoints and the determination of the true copy number values problematic. To facilitate and standardize this process, aCGH-Smooth has been developed.

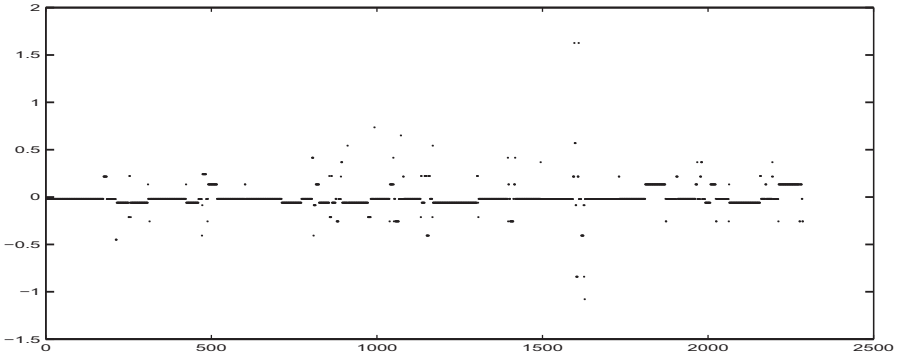
aCGH-Smooth is a tool for the automated identification of breakpoints and smoothing of micro-array comparative genomic hybridization (array CGH) data. aCGH-Smooth is written in visual C++, has a user-friendly interface including a visualization of the results and user-defined parameters adapting



(a) Tumor 2730, dots are raw data, line is result of GLS-pp



(b) Tumor 2730, dots are raw data, line is result of the manual smoothing



(c) Tumor 2730, dots are the difference between GLS-pp and the manual smoothing

Figure 4.10: Illustration comparison manual and automatic smoothing

the performance of data smoothing and breakpoint recognition. aCGH-Smooth can handle array-CGH data generated by all array-CGH platforms: BAC, PAC, cosmid, cDNA and oligo CGH arrays. The tool has been successfully applied to real-life data.

The core of aCGH-Smooth is GLS-pp as explained previously. It identifies potential breakpoints and smoothes the observed array CGH values between consecutive breakpoints to a suitable common value.

The output of aCGH-Smooth is a mapping which associates to each clone a new value. For every batch of experiments, aCGH-Smooth allows adapting the settings for smoothing and breakpoint recognition to the requirements of the raw data, depending on the biological and technical quality of the samples analyzed.

aCGH-Smooth takes as input Excel files, e.g. the ones produced by, the GenePix Pro software (Axon inc, CA), the UCSF Spot software [24] or Imagene (Biodiscovery Inc., El Segundo, CA). A file describes one array CGH experiment, which includes a sequence of clone positions in the genome and clone values measured in the experiment.

A novel option of aCGH-Smooth allows the user to set the value of a threshold used for detecting potential amplicons and outliers. The corresponding clones are not considered by the smoothing algorithm.

4.3.1 Results

We show the results of aCGH-Smooth with default parameters when applied to three types of array-CGH experiments.

Figure 4.11 shows a gastric tumor experiment, performed at the UCSF Cancer Center, with a CGH-array that has PCR representations of BAC and PAC clones as a probe spotted on the array. Each spot on the array covers around 100-200 kb of the human genome ([69]).

Figure 4.12 is a "normal to normal" experiment, except that there is an extra copy of chromosome 18. It was performed at the VU Medical Center of Amsterdam. It uses CGH-arrays that have synthetically synthesized oligos as a probe spotted on the array. Each spot on the array covers 60 bp of the human genome. As can be seen from the figure, the line is not entirely straight at all but one area, this may indicate very noisy data or a too small penalty for inserting breakpoints.

Figure 4.13 shows a "normal to normal" experiment, except that there are 3 copies of the X chromosome. It was performed at Stanford [49], and uses CGH-arrays that have cDNAs as a probe spotted on the array. Each spot on the array covers between 200 and 1500 bp of the human genome.

We found that only in few points there is a clear difference between the results given by aCGH-Smooth and by the expert. aCGH-Smooth can thus easily and effectively be applied for data generated by different analysis programs and different array-CGH platforms.

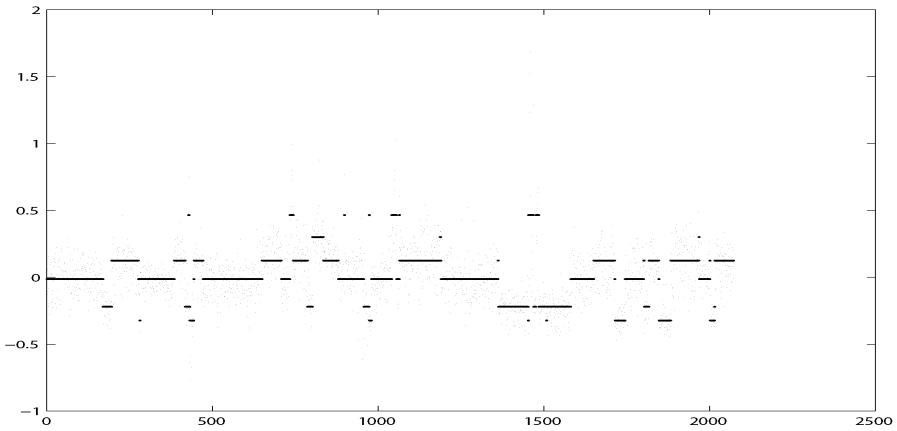


Figure 4.11: aCGH-Smooth applied to a gastric tumor

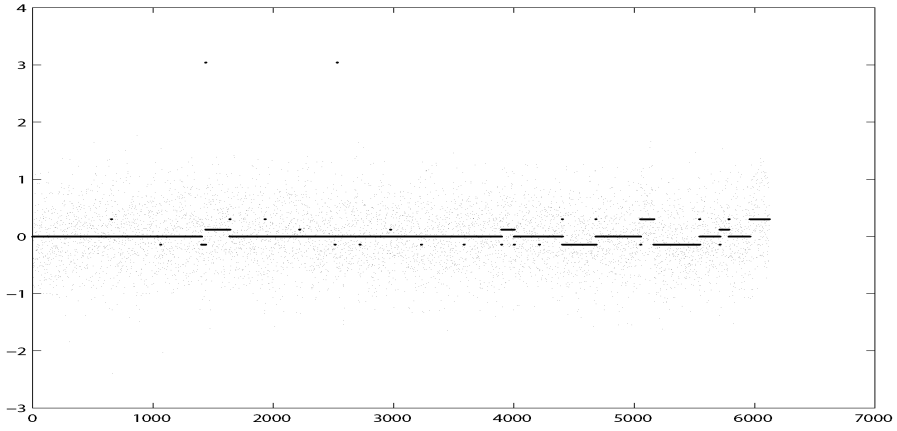


Figure 4.12: aCGH-Smooth applied to oligo data

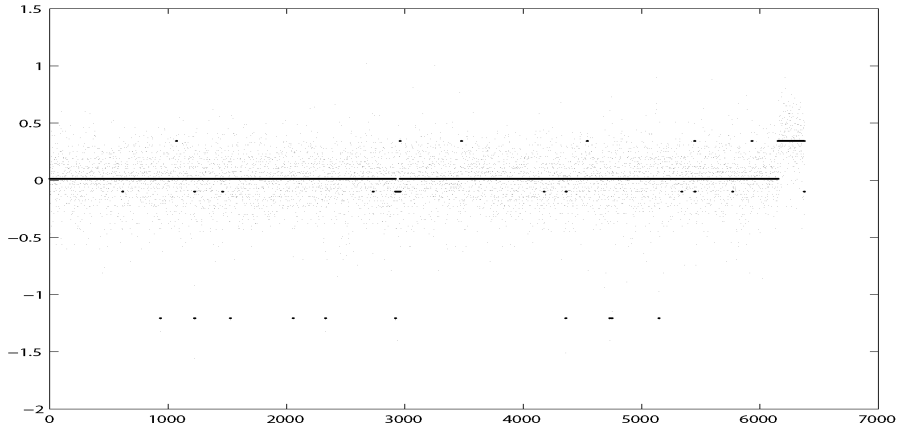


Figure 4.13: aCGH-Smooth applied to Stanford data

4.4 Related work

Other methods/tools for array-CGH analysis based on different computational approaches have been introduced at the time of or after the proposal presented in this chapter. For instance, [17] which uses Hidden Markov Models (HMM) for modeling the possible dependence of a clone with its near neighbor. In [41] a variant of the binary segmentation approach is used. In [4] a procedure that includes a moving average, k-means and dynamic programming is proposed. [67] uses a hierarchical clustering method in combination with an estimate of the false discovery rate.

Recently, an article appeared, comparing 11 different noise reduction approaches for array CGH experiments [31]. These include both segment detection methods and smoothing methods, based on diverse techniques as mixture models, Hidden Markov Models, maximum likelihood, regression, wavelets, genetic algorithms, and others. They compute the Receiver Operating Characteristic (ROC) curves using simulated data to quantify sensitivity and specificity for various levels of signal-to-noise ratio and different sizes of abnormalities. They also characterize the performance of the algorithms on chromosomal regions of interest in a real data set obtained from patients with Glioblastoma Multiforme [31]. Although no strong conclusions were drawn, the approach presented in this thesis successfully detected most of the (sometimes artificially generated) aberrations.

4.5 Future directions

The most obvious violations of the normality assumption are caused by amplifications. Results can be further improved if the clones in amplification areas are removed from the data. Amplifications are small areas on the genome with a high copy number. The genes on those areas are important candidates to be oncogenes. However, it is not easy to define an amplification unambiguously. An amplification area starts with a "big" increase of CGH value, lasts for only a "few" clones, after which the CGH values decrease "steeply". The number of clones for which an amplification "lasts at most" is a parameter. The increase and decrease of the value that are at least necessary to form an amplification depend on all value changes between consecutive clones. Say the average value change is $\overline{d}$ and the standard deviation of the value changes is s_d. Then the criterion could be $\frac{d_i - \overline{d}}{s_d} \geq T$, where T is a parameter.

Another subject that deserves attention is overlapping chromosomal positions. Since the number of spots on micro-arrays increases over time with the technological developments, an increasing number of experiments where chromosomal positions overlap can be expected. The overlapping regions may provide valuable information about the amount of noise present in the experiments.

Furthermore, it may be beneficial to gain more insight in the sources of noise of the experiments, such as cross hybridization. Incorporating this type of information into a smoothing algorithm could improve the performance. However, obtaining information about, say, the "amount" of cross hybridization may not be trivial.

From a broader perspective, noise reduced array CGH data can be valuable in combination with RNA expression array data, to detect correlations between gene expression and copy number levels.

Chapter 5

Comparing CGH platforms

5.1 Abstract

A series of studies have been published that evaluate the chromosomal copy number changes of different tumor classes using array Comparative Genomic Hybridization (array CGH), however, the chromosomal aberrations that distinguish the different tumor classes have not been fully characterized. Therefore, we used a meta-analysis of different array CGH data sets in an attempt to classify samples tested across different platforms. As opposed to RNA expression array analysis a common reference is used in dual channel CGH arrays: normal human DNA, theoretically facilitating meta-analysis. To this aim, cell line and primary cancer data sets from three different dual channel array CGH platforms obtained by four different institutes were integrated. Preprocessing methods were developed, trained and optimized using cell line data. The preprocessing performed noise reduction and transformed samples into a common format. The transformed array CGH profiles allowed perfect clustering by cell line, but importantly not by platform or institute. The same preprocessing procedures used for the cell line data were applied to 373 primary tumors profiled by array-CGH, including controls, to determine by clustering the degree of separation of different tumor classes. Results indicated that there is no apparent feature related to the institute or platform and that array-CGH allows for an unambiguous genomics meta-analysis. Major clusters with common tissue origin were identified. Tissue clusters were not pure, but interspersed by tumors of different origin, likely reflecting a limited convolution of tumor origin specific chromosomal aberrations.

5.2 Introduction

As stated in the previous chapter, array CGH is the high-resolution laboratory technique of choice for the detection of chromosomal DNA copy number

ID	Color	Inst.	Ref.	Platform	Tissue	# Exp.
1		Stanford	[49]	cDNA	Breast	44
2		UCSF	[60]	BAC	Breast	2
3		VUMC		Oligo	Breast	20
4		Cambridge	[13]	BAC	Colon	37
5		UCSF	[39]	BAC	Colon	25*
6		VUMC		BAC	Colon	20
7		VUMC		Oligo	Colon	3
8		UCSF	[59]	BAC	Fallopian	12
9		UCSF	[68]	BAC	Gastric	36
10		VUMC		BAC	Gastric	20
11		UCSF	[62]	BAC	Head and Neck	89
12		VUMC		BAC	Lymphoma	34
13		VUMC		Oligo	Lymphoma	3
14		UCSF	[65]	BAC	Prostate	4
15		VUMC		Oligo	Retinoblastoma	5
16		Stanford	[33]	cDNA	Soft tissue	25*

Table 5.1: Primary cancers. "ID" is an identifier that allows us to refer to all studies. "Inst" gives the institute that manufactured the arrays. "Ref" gives a reference for the previously published data. "Platform" provides the type of spotted probes used. The column "Tissue" provides type of tissue. "# Exp" gives the number of primary cancers collected. *) First experiments of larger set.

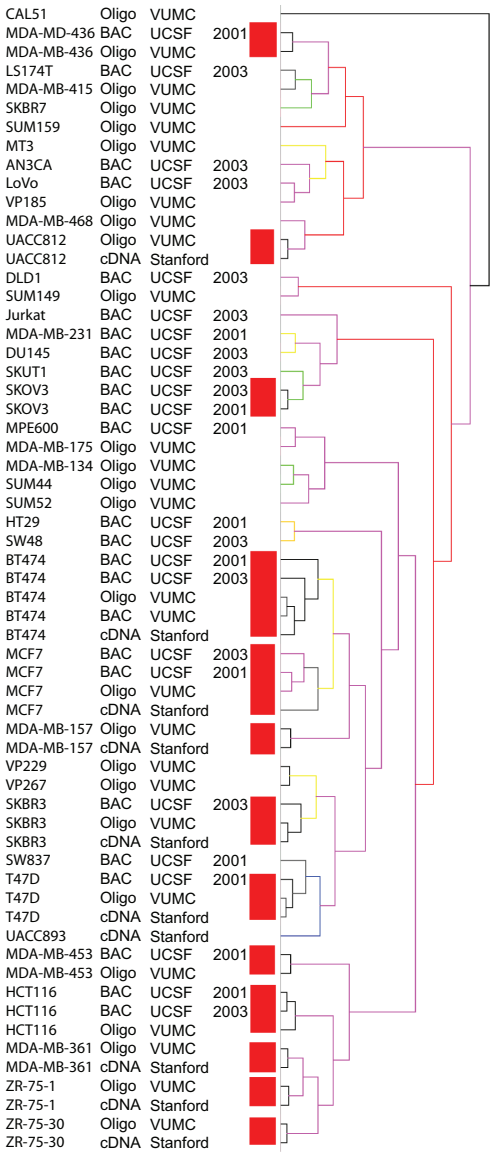


Figure 5.1: Hierarchical clustering of cell line experiments. The labels of the experiments have 4 columns. From left to right they represent the cell line, platform, institute and year. The year is only given for the UCSF platform, 2001 referring to [60] and 2003 referring to [61]. The red blocks mark experiments on the same cell line. The colors in the dendrogram indicate the support of the clusters; from complete to low support, black, gray, blue, green, yellow, orange, purple and red.

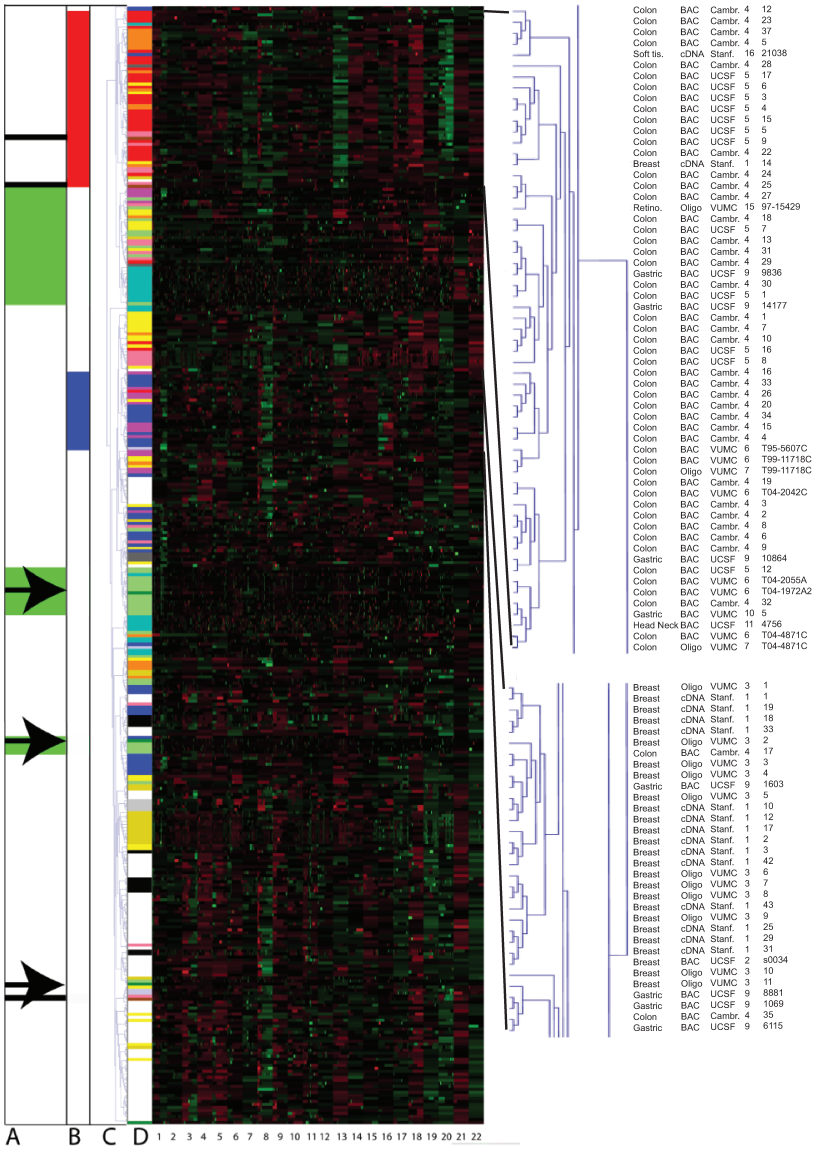


Figure 5.2: Hierarchical clustering of primary cancers. Column "A" shows the controls. The green regions are the Lymphoma clusters mentioned in the text. The arrows are the 3 Oligo lymphomas. The black bars are the three BAC and Oligo pairs that cluster together. Column "B" indicates the two large colon and breast cancer clusters. Column "C" is the dendrogram of the hierarchical clustering. The colors in column "D" correspond to the colors found in Table 5.1. It indicates the distribution of the experiments from the various sets over the clustering. To the right of the two columns is the primary cancer data. At the bottom the chromosome numbers are shown. To the right of the clustering two subclusters are highlighted. They are the main colon and breast cluster as referred to in the text. In contrast to the remainder of the clustering it shows the identifiers of each individual experiment.

aberrations (see also [46]). We and others [2] hypothesize that driver genes determine the location of aberrations. This would occur through clonal selection during the development of the cancer. Thus amplifications or other chromosomal aberrations have a certain cancer specificity, which can be exploited in a meta-analysis. For example, within a gastric cancer set we previously identified array-CGH signatures associated with different clinical outcome [68]. In addition, array CGH revealed that more advanced, aggressive or metastatic cancers often display more chromosomal aberrations [71]. This illustrates that array CGH profiles reflect tumor characteristics.

A variety of commercial and non-commercial platforms for array CGH have become available using bacterial artificial chromosomes (BACs), phage artificial chromosomes (PACs), cosmids, cDNAs, fosmids, synthetic oligo-nucleotides [46]. Several genome-wide array CGH studies, such as breast [49], fallopian tube [59], prostate [65] and soft tissue cancers [33] have been published.

In this chapter we use meta-analysis of array CGH data from different platforms and institutes. Such analysis is important since: it gives the possibility to achieve more robust and reliable results by considering data sets from different studies, it offers the ability to perform relative analysis of samples from different types of cancers, and finally it allows the identification of subgroups of samples not only within a cancer type, but also within different types of cancers. Such cross-cancer type patterns may show specific biological and / or clinical features [70]. The main advantage of meta-analysis is that whenever a new data set is generated, this data set can be compared to previously generated data sets. This will generate additional insights in the distinct features of the newly created set, which emphasizes the relevance of meta-analysis.

Array CGH data from different platforms cannot be compared directly, since each platform contains different numbers of clones at variable spacing and resolution, and the noise distributions varies across platforms [46] as well as the amplitude for a given copy number change [6]. Finally, the way each institute performs the experiments may introduce specific types of noise in the data and influence dynamic range. We developed a five-step preprocessing methodology to overcome these problems. To test and optimize the performance of the methodology we initially used cell line data obtained by different laboratories using different platforms for the preprocessing and hierarchical clustering. Subsequently, we applied the same procedure to primary cancer data from 373 samples. Array CGH data is less complex than expression array data, since the data resulting from the chromosomal copy numbers that underlie the ratios, are integers [30] and nearby chromosomal positions tend to have a highly correlated CGH value. The use of a common reference of array CGH experiments provides an intrinsic baseline, which is a big advantage for meta-analysis, compared to RNA gene expression experiments.

5.3 Material and methods

5.3.1 Data collection

We collected dual channel array CGH data sets that are publicly available from cell lines and primary cancers. To obtain a reasonable number of data sets while maintaining an overall high resolution, only those with a genome wide average resolution of 1.5 Mb or higher were selected. Some cell lines were used in experiments on different platforms at different institutes. The cell line data sets contained 7 duplicates, 3 triplicates, 1 quadruplet and one cell line was used 5 times (Table 5.2).

Dual channel array CGH profiles of 373 primary tumor samples were collected that are either publicly available or performed in our laboratories. These include either BAC arrays from the Sanger Institute in Cambridge (UK), the University of California in San Francisco (UCSF) and our own laboratory in Amsterdam (VUMC), or oligonucleotide arrays from our own laboratory or cDNA arrays from Stanford University (Table 5.1).

A large set of BAC and Oligo array experiments were performed by us and unpublished. VUMC BAC array CGH was performed according to the protocols described by [56]. VUMC oligo array CGH was performed essentially according to [9] using the Human Release 2.0 oligonucleotide library, containing 60-mer oligonucleotides representing 28830 unique genes as designed by Compugen (San Jose, CA, USA), and obtained from Sigma-Genosys (Zwijndrecht, The Netherlands). Oligo arrays were scanned using a laser scanner (Agilent Technologies, Amstelveen, NL) and analyzed using Bluefuse Software v.2.0 (Bluegnome Ltd, Cambridge, UK). For all VUMC arrays presented a reference of pooled DNA samples from 10 healthy individuals was used.

5.3.2 Preprocessing data to transform samples into a common format

Preprocessing methods were developed and applied, which perform noise reduction and transform samples into a common format.

Clone mapping The May 2004 freeze of the UCSC, ENSEMBL and CHORI databases was used. Clones not found in these databases were excluded.

Noise reduction To overcome the problem of varying noise across platforms the array CGH smoothing algorithm was applied (chapter 4 and [29]). Smoothing settings depend on the number of clones and noise profile, and therefore on platform. Previously the settings for the BAC platform were adjusted by expert opinion [30] which was applied here to the other platforms. The parameter λ in the smoothing algorithm, a value which is inversely proportional to the number of breakpoints identified, was set to 2 for the Oligo platform, 1.5 for the cDNA, and 0.8 for the BAC.

Chromosomal position sampling To deal with the varying positions of the different clones on the genome, 100 positions were sampled on each chromo-

Tissue	Cell Lines
Breast	BT474 (a,b,c,d,e), MDA-MB-231 (b), MDA-MB-453 (b,e), MPE600 (b), T47D (a,b,e), SKBR3 (a,c,e), MDA-MB-436 (b,e), CAL51 (e), MCF7 (a,b,c,e), MDA-MB-134 (e), MDA-MB-157 (a,e), MDA-MB-175 (e), MDA-MB-361 (a,e), MDA-MB-415 (e), MDA-MB-468 (e), MT3 (e), SKBR7 (e), SUM149 (e), SUM159 (e), SUM44 (e), SUM52 (e), UACC812 (a,e), VP185 (e), VP229 (e), VP267 (e), ZR-75-1 (a,e), ZR-75-30 (a,e), UACC893 (a)
Colon	HCT116 (b,c,e), HT29 (b), SW837 (b), DLD1 (c), LoVo (c), LS174T (c), SW48 (c)
Endometrium	AN3CA (c), SKUT1 (c)
Prostate	DU145 (c)
Leukemia	Jurkat (c)
Ovarian	SKOV3 (b,c)

Table 5.2: Cell line data sets. "Tissue" gives the tissue origin of the cell line. "Cell lines" gives the name of the cell line and between brackets the platforms; "a", 10 cDNA experiments performed at Stanford University [48]; "b" 11 BAC experiments performed at UCSF [60]; "c" 12 BAC experiments performed at UCSF [61]; "d" 1 new BAC experiment performed at VUMC; "e" 27 New Oligo experiments performed at VUMC.

some at equal spacing. This approach weighs each chromosome equally and emphasizes breakpoints over chromosomal length.

Interpolation The DNA copy number ratio for each sampled position was set to the ratio of the closest position in the smoothed data.

Scaling The dynamic range for the CGH ratios may vary across platforms such that a single copy gain in one platform gives a higher or lower value compared to another platform [6]. One of the most notable differences between DNA from cell lines and primary cancers is the purity of the samples. While the chromosomal DNA of cell line samples is identical in nearly 100% of its cells, cancer samples have different admixtures of cancer and normal cells. This heterogeneity of samples results in cancer / reference ratios closer to one, or equivalently log ratios closer to zero in primary cancer samples. Transforming the smoothed log ratios to z-scores was used to reduce this effect. This means that per experiment the average ratio was subtracted from all ratios and the resulting ratios were divided by the standard deviation of the original ratios.

5.3.3 Meta-analysis procedures

Cell line data

The cell line data was used to optimize the preprocessing procedure proposed in the previous section. Hierarchical clustering with the average linkage method was then used for the analysis of array CGH data [62, 68]. To overcome scaling problems we used the Spearman rank correlation distance [15, 53], since it is scale independent.

The robustness of the clustering obtained with the cell line data was evaluated with the "support tree" method in TIGR MeV [18, 53]. This is a bootstrapping method where chromosomal positions are randomly sampled with replacement and the hierarchical clustering is performed again on these data. This was done 100 times. A particular cluster that reappears frequently is likely not to be biased by a small number of clones.

Secondly, we analyzed the amount of variance due to platform or institute. This was done for each cell line and each sampled position by computing the standard deviation at that position over all the experiments on that cell line.

Primary tumor data

We applied the same preprocessing steps used with the cell line data to the primary cancer data, followed by a hierarchical clustering applied as above. The settings were kept identical to those used for the clustering applied to the cell line meta-data.

To determine the chromosomal positions that set clusters apart, the Wilcoxon rank sum test was applied. Positions were selected that have significantly different median at the corrected 0.05 p-value level with Bonferroni correction. Finally, we applied a linear support vector machine (SVM) to the significant positions. This was done using a five-fold cross validation procedure, meaning

the data set was randomly divided in five parts and to train five linear SVMs each time leaving one part out. Each time the left over part was used as test set, to estimate the generalization error. For the linear SVM the data was scaled using z-scores per experiment before selecting the positions.

5.4 Results and discussion

5.4.1 Common cell lines to optimize meta-analysis settings

Different array platforms use different type, length, mapping and number of clones, different print pins and surfaces, different labeling and hybridization procedures, different normal reference samples and DNA isolation procedures, different scanning and imaging procedures etc. This results in profiles that recognize the chromosomal aberrations, yet differ in noise distribution, amplitude and resolution (Figure 5.3).

As a first control of performance of the preprocessing steps a variance analysis was applied to the normalized meta-data set. The distribution of the standard deviations is such that for each cell line at least 75

As a second method to test the preprocessing steps for the meta-analysis we used hierarchical clustering. The dendrogram produced by the hierarchical clustering shows all experiments on the same cell line cluster together, not interspersed by other cell lines (Figure 5.1). Interestingly, we also noted that cell lines VP229 / VP 267 derived from a primary breast cancer and its relapse, respectively, cluster together in the same terminal branch of the tree. We therefore conclude that the cell line experiments cluster by copy number characteristics rather than by platform or institute.

5.4.2 Clustering of primary cancers

Hierarchical clustering was applied to the primary cancer meta-data using the same settings as used for the clustering of the cell line data. Internal controls included 3 colorectal samples that were performed independently (different days and technicians) on two different platforms, the VUMC BAC array and the VUMC oligonucleotide arrays. A second set of controls included 37 lymphoma samples, all from the VUMC pathology department, of which 34 were hybridized to the VUMC BAC arrays and 3 to the oligonucleotide arrays. In Figure 5.2 it can be seen that the 3 colon samples on the 2 different platforms perfectly pair (Figure 5.2, black bars). Lymphomas fall in two main clusters of 5 and 15 and one mixed cluster containing 8 lymphoma samples (Figure 5.2, green bars). The remaining 6 lymphoma samples are scattered throughout the meta-cluster. The three lymphomas performed on the oligonucleotide arrays (indicated by arrowheads) cluster either within one of the two main clusters and one pairs separately together with another lymphoma sample. In addition,

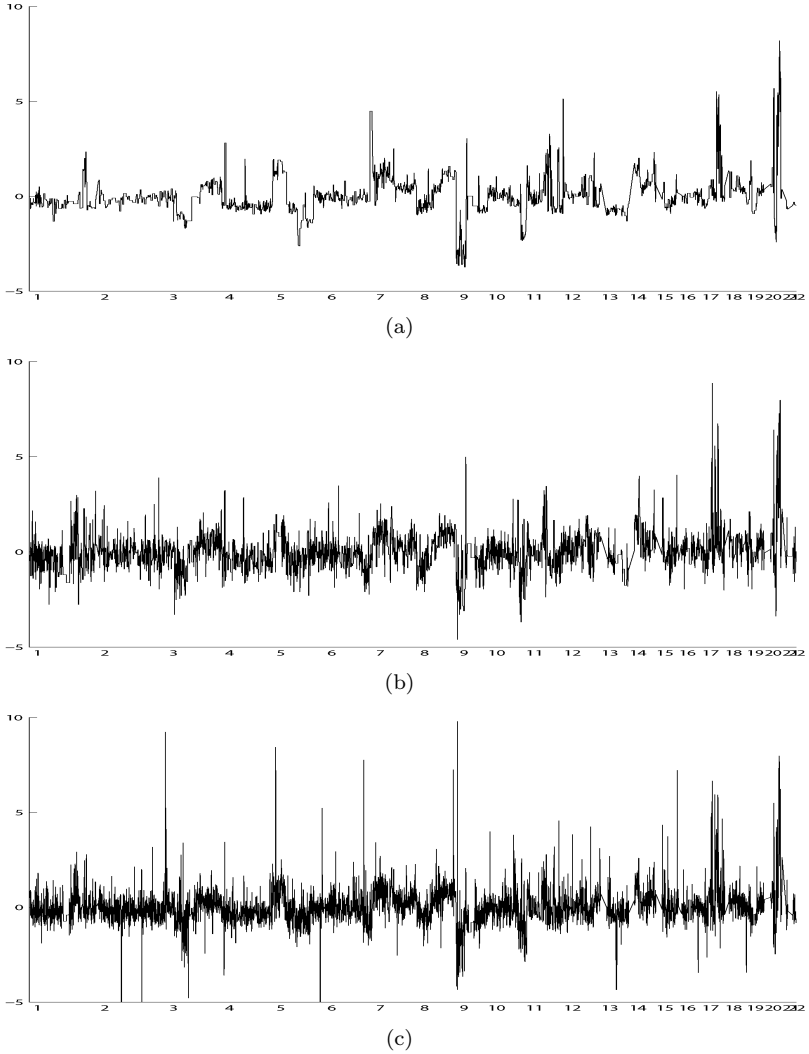


Figure 5.3: Comparison of array CGH on three different platforms by three different institutes. The figures represent three BT474 cell lines with log transformed CGH values after mapping to the same freeze, but before any other preprocessing. The x-axis represents the chromosomes. The y-axis are the log transformed CGH ratios. "a" is from [59], "b" from [48] and "c" is an Oligo experiment performed at the VUMC.

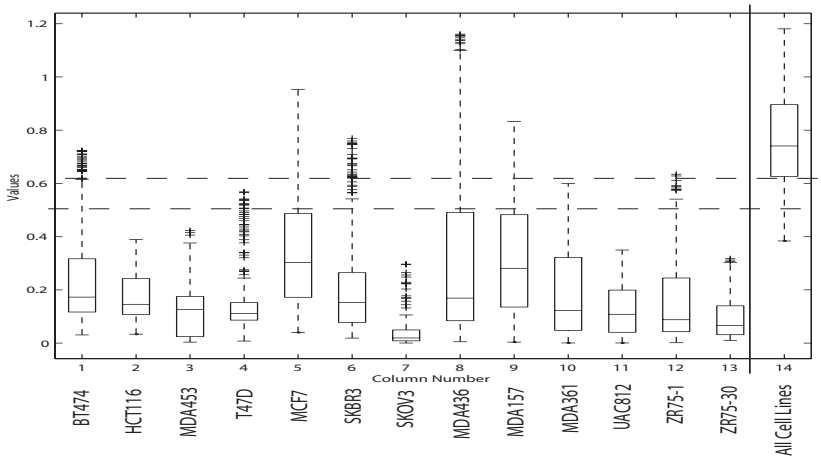


Figure 5.4: Platform dependent variation compared to variation in the different cell lines. For each cell line a boxplot of the standard deviations of all sampled positions over all experiments on that cell line is shown. The right-most boxplot thus represents the variation due to the data, since it was constructed using the entire cell line data set, whereas the other boxplots indicate the variation due to the platforms and institutes. The box has lines at the lower quartile, median, and upper quartile values. The whiskers are lines extending from each end of the box to show the extent of the rest of the data. Outliers are data with values beyond the ends of the whiskers. The whiskers end at 1.5 times the inter quartile range. The dashed lines are drawn at the lower quartile of the boxplot for the entire data set and at the largest upper quartile of the individual cell lines.

one large cluster containing the majority of colorectal samples (51) is apparent (Figure 5.2, red bar), where samples profiled on BAC arrays from three different institutes intermingle. The final proof that the meta-clustering is reliable is provided by the primary breast tumor samples, which were performed on three different platforms: cDNA, Oligo and BAC CGH arrays. One striking meta-cluster containing 24 breast tumor samples from all evaluated platforms can be recognized (Figure 5.2, blue bar).

In summary 4 controls produced the expected results. First, identical colon samples on different platforms cluster perfectly together. Secondly, the lymphoma cases show that samples from one institute and same tissue origin but profiled on different platforms cluster together. Thirdly, colon samples from different institutes on same platforms cluster together. Finally, the breast samples show that samples from different institutes and the same tissue origin cluster together. It can therefore be concluded that the meta-clustering is reliable on primary tumor data, since samples cluster by cancer type rather than by institute or platform in an unsupervised manner.

5.4.3 Analysis of large primary cancer clusters

The meta-analysis discerned major clusters with common tissue origin, the clusters are interspersed by tumors of different origin. The large colon cluster at the top is followed by the cluster with 8 lymphoma cases. Further down the largest cluster of soft tissue cancers follows and a cluster of mixed colon and gastric samples. Next cluster down is the breast tumor cluster mentioned above. Next, small breast, retinoblastoma and head and neck clusters follow. These clusters are all admixed with gastric samples. In this region we also find the largest lymphoma cluster discussed above. Some smaller clusters, including the third lymphoma cluster follow, ended by the largest gastric cluster of 11 samples and 3 bigger clusters of primarily head and neck origin.

The breast and colon clusters are the most informative clusters with respect to the performance of meta-clustering. The breast cluster is the most striking, since it represents arrays from all three platforms, BAC, oligo and cDNA and the colon cluster represents BAC arrays from three different institutes.

To determine the chromosomal positions that discriminate the breast and colon clusters a Wilcoxon rank sum test was used. Subsequently, a linear SVM was applied to the significantly different positions. The error rates indicate array CGH data can be used to discriminate between the colon and breast cancer clusters using only the positions with significant difference in median (table 5.3).

Distinct chromosomal areas emerge that discriminate the main breast cluster from the large colon cluster (Figure 5.5). These areas correspond to 6 chromosomal regions, namely: 5q35.1-2 representing 5 consecutive positions of the 100 chromosomal positions, 7p13 through 7p11.2 with 4 consecutive positions representing the 4 chromosomal bands closest to the centromere, the entire chromosome 13, a large centromeric part of chromosome16p (16p13.13 -

	Mean (Std. dev.)
Error	0.014286 (0.031944)
Sens	0.980000 (0.044721)
Spec	1.000000 (0.000000)

Table 5.3: Breast cancer cluster (negative) and colon cancer cluster (positive) separated using only significantly different chromosomal positions. 5-fold cross validated linear SVM. The first row is the error, which is the fraction of misclassifications. The second row is the sensitivity, which is the number of correctly positive classified experiments divided by the total number of experiments classified as positive. The third row is the specificity, which is the number of correctly negative classified experiments divided by the total number of experiments classified as negative. The second column shows the error, sensitivity and specificity of a 5-fold cross validated linear SVM. The reported error, sensitivity and specificity are averages over the 5 runs and between brackets are the standard deviations.

Figure 5.5: Genomic locations with significant difference in median between the breast and colon cancer clusters (179 positions), tested with the Wilcoxon test. The horizontal axis represents the chromosomes. The dots are present at genomic locations that show a significant difference in median between the groups.

16p11.2, 28 clones), the majority of chromosome 18q (q12.2 - q23, 41 clones) and 20q12 represented by only one clone. The identified chromosomal regions confirm much of the known features of these types of tumors and are consistent with many of the published array-CGH and cytogenetic studies. For example, chromosome 20 is amplified in the majority of colon tumors [13, 39, 21, 50], but only in 20% of breast tumors [2, 49]. Chromosome 7 is a typical feature of many colon tumors, but not frequently detected in breast tumors [13, 39].

In conclusion, our meta-methods can be applied to detect the differences between meta-clusters. Similarly, supervised methods can be used to detect differences between tumors or tumor subtypes whereby array CGH platforms can be intermixed, which is beyond the scope of this chapter.

5.5 Conclusions

The unsupervised meta-analysis of primary tumors reported here is the first description of array-CGH-based cancer meta-clustering. We have shown unambiguous meta-clustering of array CGH data from a selection of different platforms and institutes. One of the advantages of array CGH as a genomics array technique is that meta-analysis of array CGH is more straightforward to perform as well as less ambiguous for interpretation compared to meta-analysis of other types of genome wide data such as expression micro-arrays [37, 51]. We conclude, based on the cell line data, the lymphoma, breast and colon controls that platform effects are negligible. Figure 5.3 shows that all platforms have a distinct noise distribution. Addressing this noise issue is not trivial and we described here an algorithm that deals effectively with it.

Array CGH is a powerful technique that measures chromosomal copy number aberrations, but other genetic aberrations, such as small nucleotide polymorphisms [72], translocations, methylations [71] balanced translocations [42] and copy number variations are not taken into account [58]. The ratios that underlie array CGH thus result from chromosomal copy number aberrations only, which are integers [33] and nearby chromosomal positions tend to have a highly correlated CGH value [64]. Although some major clusters were identified, many tumors are scattered throughout, which may be explained by the algorithms used for this meta-clustering, in which emphasis is put on larger chromosomal regions, such that narrow specific amplifications, like the one for Erbb2 in breast cancer [49], may be missed.

For the meta-clustering presented we have made a choice that represents a balance between breakpoints and chromosome length and was justified by performance in the cell lines. Other approaches and choices may lead to different clustering, but that does not influence the quality of the meta-clustering as presented here.

Much new information can be gained using meta-analysis, whenever a new series of experiments are performed on whichever array CGH platform. For example, the region 5q35.1-2 has not previously been noted as particularly

important for breast or colon tumors [39], but judged from the meta-analysis seems important to be considered as an important region for either breast or colon carcinogenesis. The approach described here also opens the prospect of using array-CGH meta-data for molecular classification of cancers from the same site (for example, luminal versus basal types of breast cancer) and also for clinical correlations.

Chapter 6

Clustering micro-array data

6.1 Abstract

This chapter shows how clustering can be performed by using support vector classifiers and model selection. We introduce a heuristic method for non-parametric clustering that uses support vector classifiers for finding support vectors describing portions of clusters and uses a model selection criterion for joining these portions. Clustering is viewed as a two-class classification problem and a soft-margin support vector classifier is used for separating clusters from other points suitably sampled in the data space. The method is tested on five real life datasets, including micro-array gene expression data and array-CGH data.

6.2 Introduction

Clustering is the problem of finding structure in the data by identifying groups of objects that are more similar to each other than to objects in other groups. The notion of similarity is domain dependent, so clustering is an ill-defined problem for which many heuristic methods based on different approaches are introduced (see e.g. [8] for a recent overview of clustering methods). Support vector machine (SVM) is a powerful technique for classification and regression [66]. In this chapter we propose a method for clustering that uses a support vector classifier for finding support vectors which represent portions of clusters. The method searches for support vectors close to clusters cores. Each support vector is used to construct a (portion of a) cluster. Then an information criterion is used for merging nearby (portions of) clusters.

We view clustering as a two-class classification problem, where class A contains data elements while class B contains other points suitably generated. Initially, class A consists of all the data, and class B consists of uniform points randomly generated in the minimum hyper-rectangle containing the data. A

soft-margin support vector classifier is trained for separating A and B, and the non-bounded support vectors SV_A of class A are considered as candidate representatives of (portions of) clusters. Next, SV_A is removed from A, B is set to SV_A, and a SVM classifier is trained to separate the resulting new classes A and B. The idea is that at each iteration the non-bounded support vectors of the actual class A get closer to the cores of some clusters. At each iteration the same SVM parameters are used, with small soft-margin constant and kernel width, resulting in few non-bounded support vectors. A heuristic criterion is used for deciding when to stop the iterative process. The non-bounded support vectors selected at the end of the iterative process are used for building clusters by assigning each data point to the closest support vector. Finally, near clusters are merged using a model selection criterion.

We conducted experiments on real life data sets to test the effectiveness of this clustering method. We focused mainly on real life data sets from biological experiments, in particular micro-array gene expression data and array-CGH data. The results of the experiments are satisfactory and indicate that support vector classifiers can be used in combination with a model selection criterion for clustering.

6.3 The clustering algorithm

The clustering method employs a soft-margin SVM classifier with Gaussian kernel

$$K(x, y) = e^{-\frac{||x-y||^2}{2\sigma^2}}$$

with width parameter σ. In the input space the points for which the SVM decision function is equal to 1 describe boundaries enclosing the data. Support vectors lay on or outside these boundaries, where non-bounded support vectors [1] lay on these boundaries. The SVM parameters C (see section 3.2.2) and σ affect the form and smoothness of the boundaries. Small values of C and σ induce smooth boundaries with few non-bounded support vectors and more bounded ones. Non-bounded support vectors can be viewed as representatives of portions of clusters. Clusters can be obtained by joining these portions using a model selection criterion based on the Bayesian information criterion (BIC) [57]. According to this criterion, the most probable model is the one that maximizes the log-likelihood of the model given the observed data minus a term which takes into account model dimension.

Here a model is a clustering $Cl = \{Cl_1, \ldots, Cl_k\}$ of X, where X is the data set. We use a scoring function of the form

$$score(Cl) = -2\text{log-likelihood}(X) + \lambda N,$$

[1] a support vector is non-bounded if its Lagrange multiplier is smaller than C

where N is the number of free parameters of the model and λ is a penalty weight. (In our experiments we use $\lambda = 5$.) For the data experiments described in Table 6.1 we formed the likelihood under the assumption that the elements of each cluster are sampled from a spherical Gaussian distribution, with the mean vector depending on the cluster, but the same variance for each cluster. The smaller the score the better the model. We will compare clusterings using *score* for deciding whether to join two clusters and also to choose a clustering amongst those generated by running the clustering algorithm a number of times with different initializations of B (see Step 1 below).

A high level description of the algorithm is given in figure 6.1, where := denotes variable assignment, the variables A and B denote the two classes to be separated, R the set of support vectors selected as representatives, and Cl the final clustering. The algorithm is called FJC (Find and Join Clusters). The steps illustrated in figure 6.1 and 6.2 are explained below.

In Step 1, different sets B of uniform points can be generated depending on the random seed used, which may yield different clusterings. Thus FJC is executed a number of times and a preferred clustering is selected using *score*.

Two heuristic criteria are used in FJC: the way in which R is chosen in Step 2 and the strategy used for joining clusters portions in Step 4.

In Step 2 the biggest of the Cl_A's generated in the iterative process is selected as set of representatives of (portions of) clusters. This is a rather conservative criterion based on the intuition that a higher number of non-bounded support vectors may indicate a better separation of the classes. At each iteration the number of elements of class A decreases, so it would seem more intuitive to use the ratio (cardinality of Cl_A)/(cardinality of A). However, experiments on artificial datasets indicate that by using this latter criterion (representatives of) smaller clusters may be lost.

In Step 4 the algorithm tries to join portions of clusters. We use a simple heuristic where pairs of clusters are joined if the *score* of the resulting clustering decreases. Pairs of clusters are selected as follows: one cluster is chosen (in textual order) from the actual clustering, its mean vector is computed, and then the cluster closest to the mean vector is chosen. This heuristic has the advantage of being fast and yielding satisfactory results.

The algorithm is implemented in Matlab and uses the Matlab implementation by Gavin C. Cawley[2] of Platt SMO algorithm.

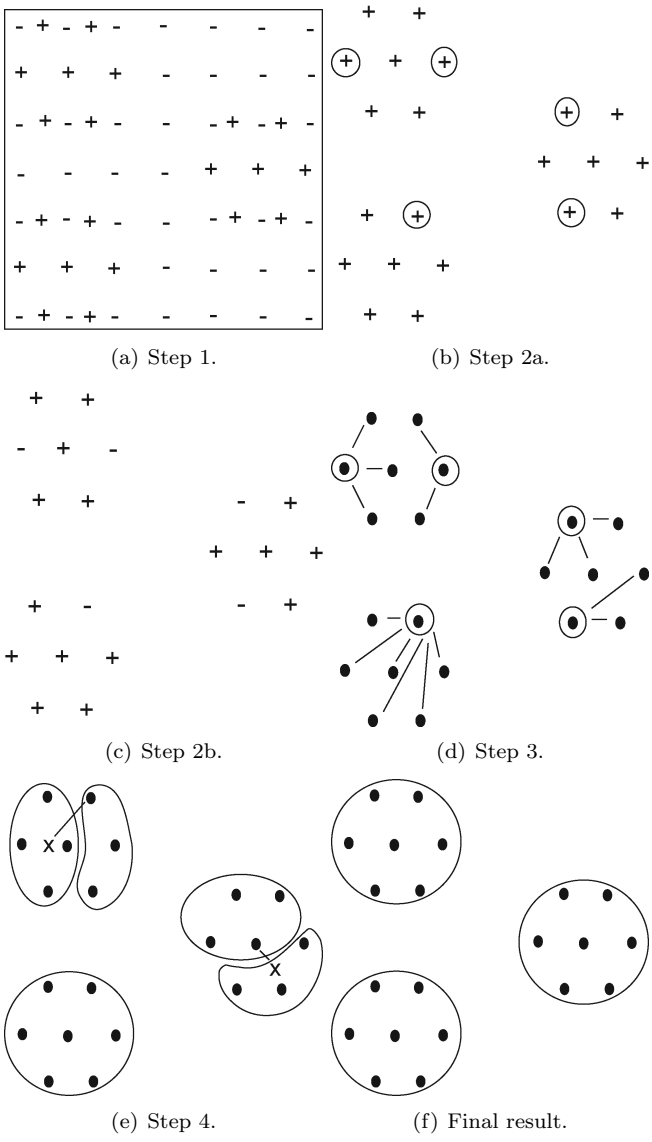
6.4 Experiments

We considered five datasets from biological experiments. The first two data sets, available at the "University of California machine learning" (UCI ML, [11]) repository, are often used as benchmarks for classification algorithms: the popular Fisher `iris` dataset with 150 points, 4 attributes, 3 classes, 50 points per class; the Wisconsin breast cancer `wdbc` with 569 points, 30 attributes,

[2]available at http://www.sys.uea.ac.uk/ gcc/svm/toolbox/

```
FJC CLUSTERING ALGORITHM
Input. X (the dataset of size n).
Output. Cl (A clustering of X).
1. Initialization.
   A := X,
   B := {n randomly generated uniform points
       in the hyper-rectangle containing X},
   R := {},
   C := C_0,
   sigma := sigma_0 (in our experiments C=2, sigma=0.1).
2. Find representatives of clusters portions.
Repeat the following two steps for n_iter
    (in our implementation n_iter=4).
   2.a Apply SVM(C_0,sigma_0) to separate A and B,
         Cl_A := {non-bounded support vectors of A},
         If (Cl_A contains more elements than R)
            then R := Cl_A.
  2.b SV_A := {the support vectors of class A},
        A := A minus SV_A,
        B := SV_A,
3. Build clusters portions.
Let R = {r_1,...,r_k} (obtained from Step 2).
Cl := {Cl_1,...,Cl_k}  with
Cl_i = {x in X closer to r_i than to any other r_j}.
4. Join clusters portions.
Repeat the following statement until Cl does not change.
   for each  Cl_i in Cl:
      c_i := mean vector of Cl_i,
      Find Cl_j containing a point closest to c_i,
      Cl_union := (Cl-{Cl_i,Cl_j}) union {Cl_i,Cl_j},
      If  (score(CL_union) < score(Cl))
            then Cl := Cl_union.
```

Figure 6.1: High level description of FJC.



(a) Step 1. (b) Step 2a.

(c) Step 2b. (d) Step 3.

(e) Step 4. (f) Final result.

Figure 6.2: Graphical description of FJC.

dataset	data	classes	clusters	nbsv	error	Jaccard	robustness
cgh	55	2	4	28	1	0.95	0.69
colon	62	2	2	54	12	0.54	0.61
wdbc	569	2	7	7	22	0.35	0.74
leukemia	72	3	3	36	3	0.91	0.77
iris	150	3	3	11	11	0.78	0.75

Table 6.1: Results of FJC on five data sets.

2 classes, 357 benign and 212 malignant. The `colon` dataset [3] consists of micro-array measurements of 2000 genes over 44 colon cancer and 22 non-cancer tissues. Here we clustered tissues, so we had 62 points, and selected the 200 genes having highest variance over the tissues. The `leukemia` dataset [16] consists of micro-array measurements of 12582 genes over 72 tissues of three Leukemia types, ALL (24), MLL (20), and AML (37). We selected the 50 genes having highest variance over the tissues. Finally the `cgh` data set consists of array CGH measurements of 1354 clones of the genome over 43 stomach cancer and 13 ovarian cancer tissues. This dataset was provided by our colleagues at the Vrije Universiteit Medical Center of Amsterdam. We selected the 150 clones having highest variance over the tissues.

We ran FJC on each dataset 10 times and selected the best clustering according to *score*. The results of the experiments are given in table 6.1, where the columns contain: the dataset name, the number of data, the number of classes (the "true" clusters), followed by the results of FJC, that is, the number of clusters, the number of non-bounded support vectors, the number of points misclassified, the Jaccard score and the robustness. The Jaccard score is $p_{11}/(p_{11}+p_{10}+p_{01})$ where p_{11} is the number of pairs of points belonging to the same cluster in both the 'true' and FJC clustering, p_{10} (respectively p_{01}) is the number of pairs that belong to the same cluster in the FJC (respectively 'true') clustering but not in the 'true' (respectively FJC) one.

The robustness gives an indication of FJC consistency in assigning pairs of points to the same or different clusters over the runs. For each pair of points x_i, x_j we compute the fraction f_{ij} of times they occur in the same cluster over the 10 runs, and then $a_{ij} = 1 - 2\text{minimum}(f_{ij}, 1 - f_{ij})$. For a pair of points, a_{ij} provides a measure of the confidence of the majority vote decision of assigning x_i and x_j to the same or different clusters. The confidence is 1 when x_i, x_j are either always or never in the same cluster. The robustness is the average confidence over all pairs of points.

The results indicate that FJC is a robust clustering method, capable of identifying "true" structure in all five data sets. The misclassification errors of FJC are comparable to those found by state-of-the-art classification methods. In particular, on the `cgh` and `leukemia` data set FJC is able to identify the classes almost perfectly.

In order to show the benefit of using the support vector classifier in FJC, we

consider the algorithm obtained from FJC by removing the SVM part. This amounts to start FJC from Step 4 with Cl consisting of one data point per cluster. The resulting algorithm yields the following results: 3 clusters, error 4 for `cgh`; 2 clusters, error 15 for `colon`; 15 clusters, error 25 for `wdbc`; 2 clusters, error 26 for `leukemia`; and 4 clusters, error 26 for `iris`.

6.5 Conclusion

This chapter showed that non-parametric clustering can be performed by using support vector classifiers and model selection. The results indicate that FJC is a robust clustering method, capable of identifying "true" structure in all five data sets. In particular, on the `cgh` and `leukemia` data set FJC is able to identify the classes almost perfectly.

6.6 Related work

The use of SVM for clustering has been proposed by Ben-Hur et al. [7]. However, they view clustering as an one-class classification problem, where the data is distinguished from the rest of the feature space [54, 55, 63, 66] by finding a sphere with minimal volume enclosing the data. The boundary of the sphere in the input space forms a set of closed contours which are interpreted as cluster boundaries, where each closed boundary forms a cluster. The authors introduce a heuristic algorithm for finding these clusters, based on the observation that two points belong to the same cluster if all the points on the line segment connecting them lie in or on the sphere in the feature space.

A method that incorporates model selection as decision test in a clustering algorithm has been introduced by Pelleg and Moore [43]. The authors propose a non-parametric clustering algorithm called X-means, where a clustering is incrementally constructed by splitting clusters and using BIC as criterion for accepting the splittings. The algorithm is an extension of K-means with efficient estimation of the number of clusters.

6.7 Future directions

We conclude with some issues we intend to address in future work. Alternative ways to initialize the class B of points in Step 1 of FJC. A possibility is finding the sphere with minimum volume in the feature space enclosing the data [63], and consider the resulting support vectors as class B and the rest of the data as class A. We intend to adapt the model selection criterion used for joining clusters. We shall investigate different methods for input dimension reduction, in order to make FJC work with high dimensional data sets.

Chapter 7

Analyzing proteomics data

7.1 Abstract

In this chapter we analyze two proteomic pattern datasets containing measurements from ovarian and prostate cancer samples. In particular, a linear and a quadratic support vector machine (SVM) are applied to the data for distinguishing between cancer and benign status. On the ovarian dataset SVM gives excellent results, while the prostate dataset seems to be a harder classification problem for SVM. The prostate dataset is further analyzed by means of an evolutionary algorithm for feature selection (EAFS) that searches for small subsets of features in order to optimize the SVM performance. In general, the subsets of features generated by EAFS vary over different runs and over different data splitting in training and hold-out sets. Nevertheless, particular features occur more frequently over all the runs. The role of these "core" features as potential tumor biomarkers deserves further study.

7.2 Introduction

Surface-enhanced laser desorption/ionization time-of-flight mass spectronomy (SELDI-TOF MS) is a recent laboratory technology which offers high-throughput protein profiling. It measures the concentration of low molecular weight peptides in complex mixtures, like serum (cf. e.g. [23]). Because it is relatively inexpensive and noninvasive, it is a promising new technology for classifying disease status. The technology is explained in more detail in section 2.4.

Given proteomic profiles for a sample of healthy and diseased individuals it is desired to build a classifier for tumor diagnostics and to identify the protein masses that are potentially involved in the disease. Because of the large number of features (the m/z ratios) and the small sample size (the specimens), the second problem is tackled using heuristic algorithms for feature selection.

In this chapter we report the analysis of two data sets obtained by applying

SELDI-TOF MS to serum samples. The first dataset concerns measurements from women with or without ovarian cancer, and was previously analyzed in [45, 35]. The second data set contains samples from patients with prostate cancer and patients with benign prostate conditions, and was analyzed in [44]. Both data sets are publicly available from the NCI/CCR and FDA/CBER Clinical Proteomics Program Databank (http://clinicalproteomics.steem.com/).

As preliminary analysis we first investigated the extent to which single m/z ratios can be used to discriminate the two classes of healthy and cancer state samples. Secondly we report the error rate of support vector machine (SVM) classifiers using the full protein profiles. It turns out that the ovarian cancer dataset is "easy" for a linear SVM classifier, whereas the prostate cancer dataset is "harder".

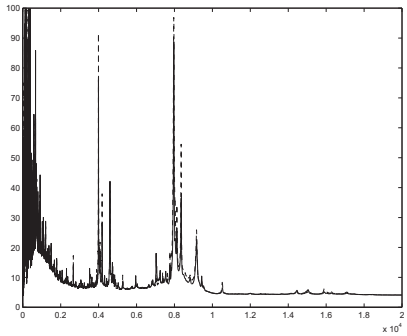
We performed a further analysis of the prostate cancer dataset by means of a feature selection algorithm based on EAs. We introduce an EA for feature selection, called EAFS (Evolutionary Algorithm for Feature Selection), in order to identify small subsets of features that discriminate the healthy and cancer groups. The results over multiple data splittings (into training, test and validation set) and multiple EA runs show that the method is slightly unstable. However, specific features occur most frequently in the solutions of multiple runs. Further study is needed in order to assess the role of these "core" features as potential tumor biomarkers.

7.3 Data analysis with all features

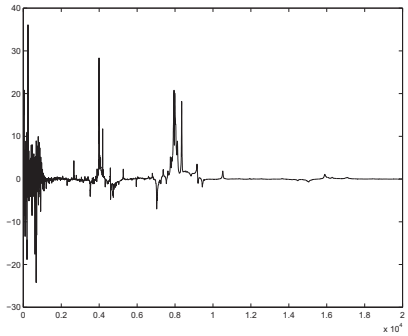
The "ovarian dataset" (8-7-02) consists of 253 samples, with 91 controls and 162 ovarian cancers, which include early stage cancer samples. The "prostate dataset" contains 322 samples, with 69 cancers and 253 healthy (or benign) samples.

We analyzed the two datasets in order to assess how difficult it is to separate healthy and cancer groups. Figure 7.1 shows properties of the mean values of the two classes for the ovarian dataset. Parts (a) and (b) of the figure indicate that the healthy and cancer classes differ only in a few regions substantially in mean. Because the variances in the two samples vary significantly with m/z ratio, the t-test applied for each m/z ratio separately is nevertheless significant for a much larger number of m/z ratios. In fact, as shown in part (c), there are many p-values equal to zero all across the full range of m/z-values, and "most" p-values are close to zero, as shown in part (d), which is a histogram with 100 bins. This seems to suggest that it is not difficult to find a good classifier for the ovarian data set. The same information on the prostate data set is given in Figure 7.2. We can see there are fewer features with significant difference in mean in the prostate than in the ovarian dataset (part (d) of the figures). Thus finding a good classifier for the prostate dataset seems to be a more difficult task.

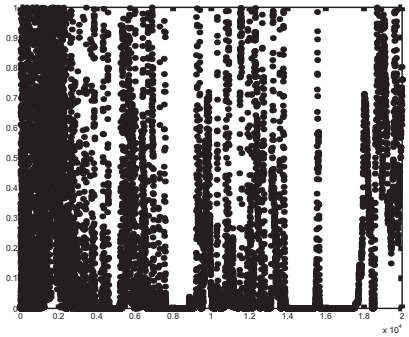
We classify the data using all the features and a SVM classifier. The choice



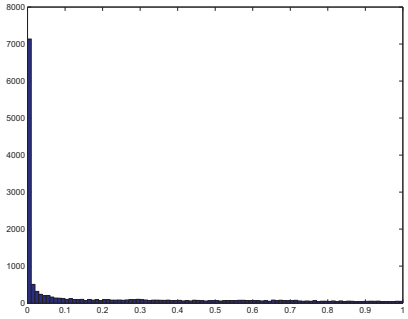
(a) Means of cancer (solid) and healthy (dashed) classes



(b) Mean healthy minus mean cancer class

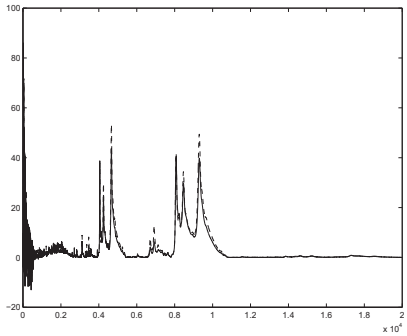


(c) p-values of t-test between groups

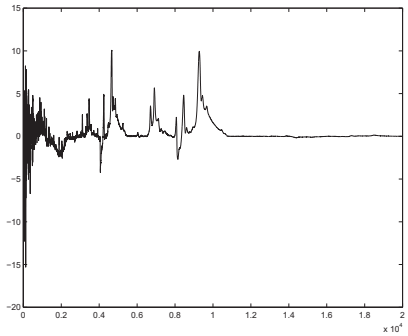


(d) Number of p-values

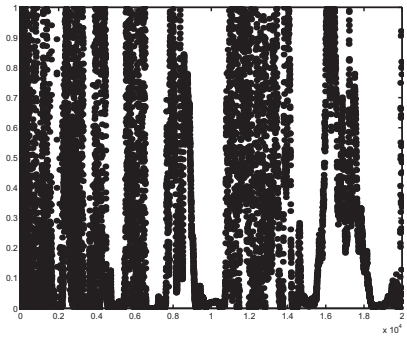
Figure 7.1: Mean values analysis of ovarian cancer data set.



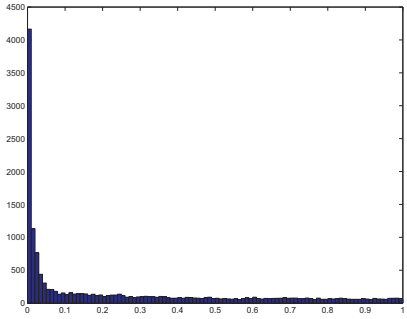
(a) Means of cancer (solid) and healthy (dashed) classes



(b) Mean healthy minus mean cancer class



(c) p-values of t-test between groups



(d) Number of p-values

Figure 7.2: Mean values analysis of prostate cancer data set.

	Lin., Over.	Quad., Ovar.	Lin. Pros.	Quad. Pros.
Test Error	**0.0028**	0.0040	0.0600	**0.0590**
	(0.0061)	(0.0065)	(0.0201)	(0.0246)
Sens.	0.9987	**0.9988**	0.8884	**0.8936**
	(0.0044)	(0.0042)	(0.0596)	(0.0719)
Spec.	**0.9947**	0.9912	0.9539	**0.9552**
	(0.0153)	(0.0169)	(0.0205)	(0.0261)
Pos. Pred.	**0.9969**	0.9951	0.8361	**0.8475**
Val.	(0.0090)	(0.0097)	(0.0630)	(0.0867)

Table 7.1: Average results over 25 runs of SVM with linear and quadratic kernel, on ovarian and prostate data.

of SVM is motivated by its good performance also in the presence of many features. In our experiments we use soft-margin SVM classifiers with $C = 100$, and two types of kernel functions, linear and quadratic.

Table 7.1 contains the results obtained by applying an SVM classifier with linear and quadratic kernel to the ovarian and prostate cancer data sets. Experiments are conducted using the SVM software package LibSVM. We perform 25 runs, where in each run the data is randomly partitioned into training (60%) and test (40 %) sets, a classifier is induced by the SVM learning algorithm applied to the training set, and its classification error rate on the test set is measured. Each entry of the table contains the average result over 25 runs (with standard deviation written between brackets) for a specific pair of SVM classifier and data set.

In all the runs the SVM classification error on the training set is zero. The table contains the average test error rate, sensitivity (number of cancer samples correctly classified divided by total number of cancer samples), specificity (number of healthy samples correctly classified divided by total number of healthy samples), and positive predictive value (number of cancer samples correctly classified divided by total number of samples classified as cancer).

On the ovarian dataset, the SVM classifier with linear kernel gives excellent results, being able to output the correct diagnosis in many runs, in particular cancer samples are almost always detected (pred. pos. val. and sensitivity are close to 1). Thus the ovarian dataset can be considered easy for a linear SVM classifier. Other methods applied to this dataset obtain also very good results. In [45] a commercial package that uses a genetic algorithm (GA) based feature selection method is applied. The GA searches in the space of all subsets of features. It uses a fitness function that scores a feature subset according to its ability to cluster samples in consistent groups, that is, groups containing samples with equal class. The authors report almost perfect classification for a specific data splitting in training and test sets. The paper does not mention results obtained by cross-validation. A more thorough analysis of the ovarian

data set is performed in [35], where 10-fold cross validation is applied, and different classification and feature selection methods are considered. Features are first "smoothed" by means of a discretization algorithm. Perfect classification is achieved using an SVM classifier with quadratic kernel, when all features are used but also when a small subset of 17 features is used. This feature subset is generated using a feature selection algorithm that iteratively constructs sets of features using the best-first-search and a scoring criterion, for selecting a best feature subset, which considers the correlation between pairs of features and between feature and class.

On the prostate dataset, sensitivity and predicted positive values are lower than specificity, possibly due to the unbalanced distribution of the two classes, where cancer samples are about 1/3 of the healthy ones. The results indicate that the prostate data set is somewhat harder to classify than the ovarian one, when using SVM with a linear or quadratic kernel and all the features. In [44] the GA-based commercial package described above is applied to this data set. The authors identify a subset of 7 features that allow their classification method to obtain 0.95 sensitivity, 0.78 specificity, and 0.1992 test error rate.

In summary, on the ovarian dataset a soft-margin SVM linear classifier provides a good diagnostic tool, while for the prostate dataset the sensitivity achieved is still too low hence does not allow a direct use of this classifier in diagnostics. An early stage tumor diagnostic tool should have sensitivity equal to 1 and specificity very close to 1.

7.4 An EA-based method for feature selection

In this section we describe a novel method for feature selection based on evolutionary algorithms (EA). Given a dataset and a learning algorithm, the goal of feature selection is to find a "small" subset of features that minimizes the generalization error (that is, the classification error on new examples) of the classifier induced by the learning algorithm when run only on the selected features.

The data is randomly partitioned into a training, a test and a validation sets (in the experiments these sets contain 60%, 30% and 10% of the data, respectively). The training and test sets are used in the feature selection algorithm and the validation set is used for assessing the performance of the resulting classifier on new data. In the standard wrapper model for feature selection, one searches for a feature subset that minimizes the test error of the classifier trained on data restricted to that feature subset. In [40], Ng shows that the main source of error in standard wrapper algorithms, when many irrelevant features are present, comes from over-fitting hold-out or cross-validation data. He proposes an exact algorithm which is more tolerant to the presence of many irrelevant features. The algorithm, called ordered-fs, works in two phases. First, for each feature set size $i \in [1, m]$, where m is the maximum number of features permitted, the algorithm finds a feature set of size i that

```
function EAFS
{
 generate initial populations
 while (termination criterion not satisfied)
 {
  select a population
  select two parents from that population
  generate offspring using uniform crossover
  apply mutation to offspring
  find populations with right number of features
  replace worst individuals with offspring
  determine fitness (error SVM on training set)
  if (offspring has very good fitness)
    apply migration operator
 }
}
```

Figure 7.3: Outline EAFS.

minimizes the classifier training error. Next, the resulting m classifiers are run on the test set, and the one yielding minimum error is chosen.

The EA-based method we propose, called EAFS, is inspired by the ordered-fs algorithm. The core of EAFS (illustrated in figure 7.3) consists of an EA which evolves a number of populations, where each population consists of individuals representing feature subsets of a given size. The populations interact by means of highly fit individuals which are used as seed for generating new individuals of other populations. Genetic operators are used for moving in the search space in order to minimize the SVM training error. At the end of the evolutionary process, the best SVM classifier of each population is run on the test set and the one yielding minimum error on this set is selected. Thus the test set is used only to determine the optimal size of the feature set. The selection of an optimal feature set of a given size is based only on the training set.

Feature subsets are represented by bit strings of length equal to the total number of features. A bit value equal to 1 means the corresponding feature is considered by the learning algorithm, while a 0 means it is discarded. Individuals of each population are initialized by means of n-tournament selection which uses a feature ranking obtained from t-tests on all single features. The fitness of a feature subset F is equal to the training error of the SVM classifier restricted to the features of F. Mutation removes a feature from F and adds a new one, where both features are randomly selected. Standard GA uniform crossover is used. While mutation does not affect the size of a feature subset, this is not the case for crossover. Thus if the feature set size of an offspring is different from the one of its parents, it migrates to another population. At each

	Training	Test	Validation
Error	0.0617 (0.0254)	0.0880 (0.0313)	0.1116 (0.0515)
Sensitivity	0.7853 (0.0926)	0.7037 (0.1193)	0.6315 (0.2069)
Specificity	0.9827 (0.0118)	0.9658 (0.0238)	0.9484 (0.0456)
Pos. Pred. Val.	0.9309 (0.0451)	0.8437 (0.0957)	0.7424 (0.2178)

Table 7.2: Results of EAFS with linear SVM

	Training	Test	Validation
Error	0.0463 (0.0287)	0.0774 (0.0283)	0.1096 (0.0579)
Sensitivity	0.8360 (0.1096)	0.7502 (0.1249)	0.6779 (0.2177)
Specificity	0.9874 (0.0109)	0.9674 (0.0216)	0.9441 (0.0525)
Pos. Pred. Val.	0.9502 (0.0431)	0.8547 (0.0948)	0.7671 (0.1936)

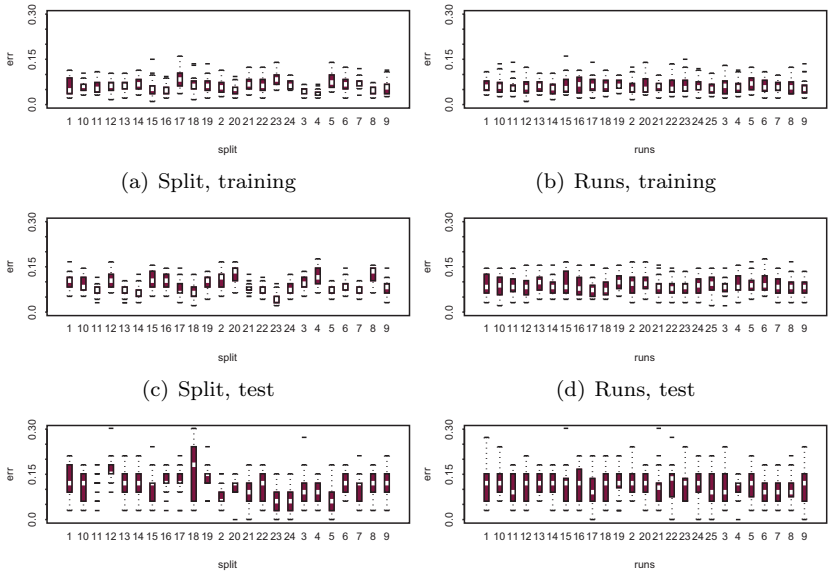
Table 7.3: Results of EAFS with quadratic SVM

iteration of the EA, a population is selected and used to generate two offspring. The EA uses tournament selection and a steady state replacement mechanism, where offspring replace the worse individuals of the population. When the EA terminates its execution, the best individual of each population is chosen and the one yielding the lowest error on the test set provides the output. In the sequel, we focus only on the application of EAFS to the prostate dataset, and will not compare EAFS to other EA-based feature selection algorithms.

7.5 Results

We used the following experimental setup: 20 populations, each one consisting of 10 individuals, 600 iterations, tournament selection of size 5, and crossover and mutation rate of 0.95. These values have been chosen after a small number of runs (using only training and test sets). We consider 24 random splitting of the data set in training (60%), test (30%) and validation (10%) sets. For each "split" of the data we run EAFS 25 times. This amounts to a total of 600 runs. Table 7.2 gives the results of EAFS with linear kernel. The values are the averages over all the 600 runs. Standard deviation is reported between brackets. Table 7.3 contains the results using a quadratic kernel. EAFS with a quadratic kernel SVM achieves best performance, but obtains sensitivity lower than that of SVM with all the features. However, a fair comparison is not possible due to the different cross validation approaches used.

In order to investigate whether the data "split" influences the performance of EAFS significantly, we perform a one-way Analysis of Variance to compare the 24 samples of 25 validation errors resulting from the 24 "splits". The difference is statistically significant, with zero p-values for both the linear and quadratic case. The "splits" explain about 24% of the variance in the validation



(a) Split, training (b) Runs, training

(c) Split, test (d) Runs, test

(e) Split, validation (f) Split, validation

Figure 7.4: Boxplot of training, test and validation errors (top to bottom), organized by split (left) and run (right).

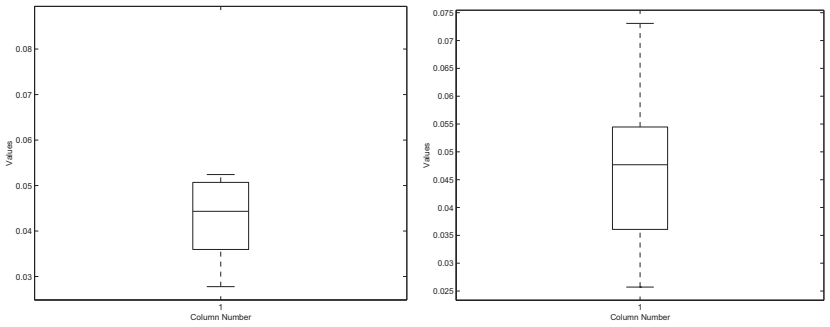
errors in the linear case and about 35% in the quadratic case, indicating that 76% and 65% is due to the randomness inherent in EAFS. A correction for the effect of "split" on the error standard deviations in Tables 7.2 and 7.3 would reduce these somewhat, but not substantially (e.g. 0.0515 becomes 0.045). Figure 7.4 gives a visual impression of the variation due to the splitting and EAFS. The three graphs show boxplots of the training, test and validation errors of the 600 runs (top to bottom), organized by "split" (left) or "run".

An important aspect of these graphs is summarized in the Figure 7.5, which shows boxplots of the standard deviations of the 24 samples of validation errors corresponding to the 24 "splits". The linear SVM suffers from one extreme split, but is otherwise more stable across relative to the splitting of the data.

We analyze now the features obtained in all the runs.

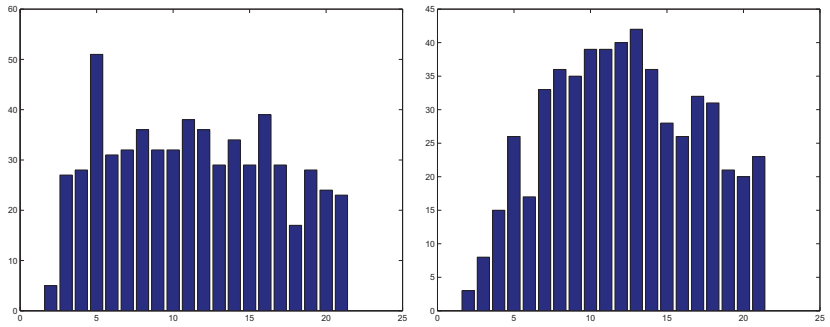
Figure 7.6 shows histograms of the final feature set sizes found over the 600 runs. The algorithm shows no preference for the largest possible feature size. We can see that EAFS with linear SVM has a peak on feature size 5, while EAFS with quadratic SVM prefers somewhat bigger feature sizes, with a peak on feature size 13.

Over all the runs EAFS with linear SVM finds 3935 features, while with quadratic SVM finds 3797 features. Figure 7.7 shows histograms of the features occurring in the solutions found over the 600 runs. It is clear that EAFS is



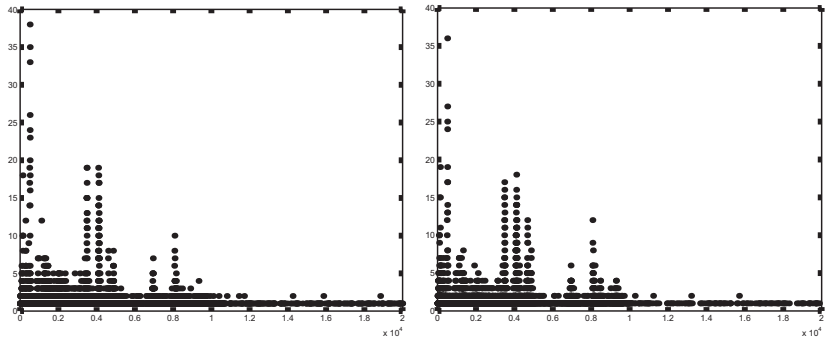
(a) Linear Kernel (b) Quadratic Kernel

Figure 7.5: Boxplots of standard deviation within 24 runs.



(a) Linear Kernel (b) Quadratic Kernel

Figure 7.6: Histograms of obtained feature set sizes.



(a) Linear Kernel (b) Quadratic Kernel

Figure 7.7: Histograms of number of feature occurrence in final solutions.

not stable, yet the histograms indicate the presence of few frequency peaks. By considering features occurring in these peaks, 47 features occurring in at least 10 runs are extracted. We perform 100 runs of the linear SVM restricted to the 47 features, with randomly chosen training and test set, and obtain 0.93 sensitivity (0.058 standard deviation) and 0.98 specificity (0.016 standard deviation). These significantly better results may be due to the fact that the selection of these features implicitly uses (almost) the entire dataset.

7.6 Conclusion

This chapter analyzed two proteomic pattern datasets. We applied SVM classifiers for tumor diagnostics, and used them in EAFS, an EA-based feature selection algorithm for the identification of potential tumor markers identification. The results do not allow us to draw strong conclusions. However, on the ovarian data set SVM with all the features exhibits excellent performance, while on the prostate data set it obtains relatively low sensitivity. Results of EAFS show that its performance on the prostate data set depends on the data splitting and EA run. Moreover, feature subsets generated by EAFS vary per run, with a small core of features occurring more often. This latter phenomenon was observed to happen also in the other methods discussed in this chapter.

Acknowledgment We would like to thank Guus Smit and Connie Jimenez from the Department of Biology of the Vrije Universiteit Amsterdam for helpful discussions on the subject of this chapter.

7.7 Future work

Future work includes the incorporation of a pre-processing phase into EAFS; the investigation of other types of classifiers; the use of knowledge-based mutation operators; and the use of multiple EAFS's runs with different splitting of training and test sets for extracting a "core" set of features from the resulting EAFS's solutions.

Chapter 8

Conclusions

Although the individual chapters have their specific results and conclusions, a higher level conclusion from this thesis is presented here.

The recent technological developments enable biologists to perform large scale measurements at various "information levels" in the human cell. Although these technologies provide interesting new opportunities, there are problems to deal with, such as experimental noise and incomparable experiments due to differences in the technique used at different institutes. In chapter 4 multiple algorithms are compared to optimize a likelihood criterion to reduce the noise in array CGH experiments. The finally chosen algorithm obtains results similar to expert opinion. A user-friendly software tool was developed to make this algorithm available to the array CGH community. Chapter 5 deals with the incomparability of the different dual channel array CGH platforms. Next to developing a preprocessing procedure to make the experiments comparable, this chapter makes preliminary investigations to identify relevant chromosomal areas to distinguish different tumor types. Much new information may be gained using meta-analysis, whenever a new series of experiments are performed on whichever array CGH platform.

On the RNA level we showed that non-parametric clustering can be performed by using support vector classifiers and model selection. The results indicate that our method is a robust clustering method, capable of identifying "true" structure in all used data sets. The misclassification errors are comparable to those found by state-of-the-art classification methods. This is explained in chapter 6.

Chapter 7 deals with data on the protein level. We applied SVM classifiers for tumor diagnostics, and used them in an EA-based feature selection algorithm for the identification of potential tumor markers identification. A series of experiments indicates that a relatively small part of the SELDI-TOF spectrum emerges as important to distinguish healthy persons from those diagnosed to have cancer.

Reproducibility and reliability of results are two issues in the analysis of

gene expression and pattern proteomic data [36, 22]. For instance, overoptimistic prognostic results are sometimes published [37]. Modern techniques for proteomics, such as SELDI-TOF, have not yet fully matured and sometimes cause a controversy [5, 34]. However, this issue is less problematic for array CGH, since a common reference (normal human cells) is used, which makes the data much easier to interpret.

Bibliography

[1] H. Akaike. Information theory and an extension of the maximum likelihood principle. In B. Petrox and F. Caski, editors, *Second International Symposium on Information Theory*, page 267, 1973.

[2] D.G. Albertson, B. Ylstra, R. Segraves, C. Collins andS.H. Dairkee, D. Kowbel, W.L. Kuo, J.W. Gray, and D. Pinkel. Quantitative mapping of amplicon structure by array cgh identifies cyp24 as a candidate oncogene. *Nat. Genet.*, 25:144–146, 2000.

[3] U. Alon. Broad patterns of gene expression revealed by clustering analysis of tumor and normal colon cancer tissues probed by oligonucleotide arrays. *Cell Biology*, 96:6745–6750, 1999.

[4] R. Autio, S. Hautaniemi, P. Kauraniemi, O. Yli-Harja, J. Astola, M. Wolf, and A. Kallioniemi. Cgh-plotter: Matlab toolbox for cgh-data analysis. *Bioinformatics*, 19:1714–1715, 2003.

[5] K.A. Baggerly, J.S. Morris, S.R. Edmonson, and K.R. Coombes. Signal in noise: evaluating reported reproducibility of serum proteomic tests for ovarian cancer. *J Natl Cancer Inst*, 97:307–309, 2005.

[6] M.T. Barrett, A. Scheffer, A. Ben-Dor, N. Sampas, D. Lipson, R. Kincaid, P. Tsang, B. Curry, K. Baird, P.S. Meltzer, Z. Yakhini, L. Bruhn, and S. Laderman. Comparative genomic hybridization using oligonucleotide microarrays and total genomic dna. *PNAS*, 101:17765–17770, 2004.

[7] A. Ben-Hur, D. Horn, H.T. Siegelmann, and V. Vapnik. Support vector clustering. *Journal of Machine Learning Research*, 2:125–137, 2001.

[8] P. Berkhin. Survey of clustering data mining techniques, 2002. http://www.accrue.com/products/researchpapers.html.

[9] B. Carvalho, E. Ouwerkerk, G.A. Meijer, and B. Ylstra. High resolution microarray comparative genomic hybridisation analysis using spotted oligonucleotides. *J Clin Pathol*, 57:644–646, 2004.

[10] N. Cristiani and J. Shawe-Taylor. *Support vector machines, and other kernel-based learning methods*. University press Cambridge, 2000.

[11] C.L. Blake D.J. Newman, S. Hettich and C.J. Merz. UCI repository of machine learning databases, 1998.

[12] R. Doll and A.B. Hill. Smoking and carcinoma of the lung. *British Medical Journal*, 2:739–748, 1950.

[13] E.J. Douglas, H. Fiegler, A. Rowan, S. Halford, D.C. Bicknell, W. Bodmer, I.P.M. Tomlinson, and N.P. Carter. Array comparative genomic hybridization analysis of colorectal cancer cell lines and primary carcinomas. *Cancer Research*, 64:4817–4825, 2004.

[14] A.E. Eiben and J.E. Smith. *Introduction to Evolutionary Computing*. Springer, 2003.

[15] M.B. Eisen, P.T. Spellman, P.O. Browndagger, and D. Botstein. Cluster analysis and display of genome-wide expression patterns. *PNAS*, 95:14863–14868, 1998.

[16] T.R. Golub et al. Molecular classification of cancer: Class discovery and class prediction by gene expression. *Science*, 286:531–537, 1999.

[17] J. Fridlyand, A.M. Snijders, D. Pinkel, D.G. Albertson, and A.N. Jain. Understanding array cgh data. *J. Multivariate Analysis*, 2004. in press.

[18] D. Graur and W.H. Li. *Fundamentals of Molecular Evolution*. Sinauer Associates, 2000.

[19] J.M. Hall, M.K. Lee, B. Newman, J.E. Morrow, L.A. Anderson, B. Huey, and M.C. King. Linkage of early-onset familial breast cancer to chromosome 17q21. *Science*, 250:1684–1689, 1990.

[20] S. Haykin. *Neural networks, a comprehensive foundation*. Prentice Hall, 1999.

[21] M. Hermsen, C. Postma, J. Baak, M. Weiss, A. Rapallo, A. Sciutto, G. Roemen, J.W. Arends, R. Williams, W. Giaretti, A. De Goeij, and G. Meijer. Colorectal adenoma to carcinoma progression follows multiple pathways of chromosomal instability. *Gastroenterology*, 123:1109–1119, 2002.

[22] J.P. Ioannidis. Microarrays and molecular research: noise discovery? *Lancet*, 365:454–455, 2005.

[23] H.J. Issaq. Seldi-tof ms for diagnostic proteomics. *Anal. Chem.*, 75:148–155, 2003.

[24] A. N. Jain, T. A. Tokuyasu, A. M. Snijders, R. Segraves, D. G. Albertson, and D. Pinkel. Fully automatic quantification of microarray image data. *Genome research*, 12:325–332, 2002.

[25] A. Jemal, T. Murray, E. Ward, A. Samuels, R.C. Tiwari, A. Ghafoor, E.J. Feuer, and M.J. Thun. Cancer statistics, 2005. *Cancer J Clin*, 55:10–30, 2005.

[26] K. Jong, E. Marchiori, M. Sebag, and A. van der Vaart. Feature selection in proteomic pattern data with support vector machines. In *CIBCB*, pages 41–48, 2004.

[27] K. Jong, E. Marchiori, and A. van der Vaart. Finding clusters using support vector classifiers. In *European Symposium on Artificial Neural Networks*, pages 223–228, 2003.

[28] K. Jong, E. Marchiori, and A. van der Vaart. Analysis of proteomic pattern data for cancer detection. In *Applications of Evolutionary Computing. EvoBIO: Evolutionary Computation and Bioinformatics*, pages 41–51, 2004.

[29] K. Jong, E. Marchiori, A. van der Vaart, and B. Ylstra. Automatic breakpoint identification and smoothing of array comparative genomic hybridization data. *Bioinformatics*, 20:3636–3637, 2004.

[30] K. Jong, E. Marchiori, A. van der Vaart, B. Ylstra, G. Meijer, and M. Weiss. Chromosomal breakpoint detection in human cancer. In *Applications of Evolutionary Computing. EvoBIO: Evolutionary Computation and Bioinformatics*, pages 54–65, 2003.

[31] W.R. Lai, M.D. Johnson, R. Kucherlapati, and P.J. Park. Comparative analysis of algorithms for identifying amplifications and deletions in array cgh data. *Bioinformatics*, 21:3763–3770, 2005.

[32] B. Lewin. *Genes VIII*. Prentice Hall, 2003.

[33] S.C. Linn, R.B. West, J.R. Pollack, S. Zhu, T. Hernandez-Boussard, T.O. Nielsen, B.P. Rubin, R. Patel, J.R. Goldblum, D. Siegmund, D. Botstein, P.O. Brown, C.B. Gilks, and M. van de Rijn. Gene expression patterns and gene copy number changes in dermatofibrosarcoma protuberans. *American Journal of Pathology*, 163:2383–2395, 2003.

[34] L.A. Liotta, M. Lowenthal, A. Mehta, T.P. Conrads, T.D. Veenstra, D.A. Fishman, and E.F. Petricoin. Importance of communication between producers and consumers of publicly available experimental data. *J Natl Cancer Inst*, 97:310–314, 2005.

[35] H. Liu, J. Li, and L. Wong. A comparative study on feature selection and classification methods using gene expression profiles and proteomic patterns. *Genome Informatics*, 13:51–60, 2002.

[36] E. Marshall. Getting the noise out of gene arrays. *Science*, 306:630–631, 2004.

[37] S. Michiels, S. Koscielny, and C. Hill. Prediction of cancer outcome with microarrays: a multiple random validation strategy. *Lancet*, 365:488–492, 2005.

[38] P. Moscato. On evolution, search, optimization, genetic algorithms and martial arts: Towards memetic algorithms. Technical report, Caltech Concurrent Computation Program, Californian Institute of Technology, U.S.A., TR No790 1989.

[39] K. Nakao, K.R. Mehta, J. Fridlyand, D.H. Moore, A.N. Jain, A. Lafuente, J.W. Wiencke, J.P. Terdiman, and F.M. Waldman. High-resolution analysis of dna copy number alterations in colorectal cancer by array-based comparative genomic hybridization. *Carcinogenesis*, 25:1345–1357, 2004.

[40] A.Y. Ng. On feature selection: learning with exponentially many irrelevant features as training examples. In *Proc. 15th International Conf. on Machine Learning*, pages 404–412, 1998.

[41] A.B. Olshen and E.S. Venkatraman. Change-point analysis of array-based comparative genomic hybridization data. In *Proc. of Joint Statistical Meetings*, pages 2530–2535, 2002.

[42] A. Oostlander, G. Meijer, and B. Ylstra. Microarray-based comparative genomic hybridization and its applications in human genetics. *Clin Genet.*, 66:488–495, 2004.

[43] D. Pelleg and A. Moore. X-means: Extending K-means with efficient estimation of the number of clusters. In *Proc. 17th International Conf. on Machine Learning*, pages 727–734. Morgan Kaufmann, San Francisco, CA, 2000.

[44] E.F. Petricoin. Serum proteomic patterns for detection of prostate cancer. *Journal of the National Cancer Institute*, 94:1576–1578, 2002.

[45] E.F. Petricoin. Use of proteomic patterns in serum to identify ovarian cancer. *The Lancet*, 359:572–577, 2002.

[46] D. Pinkel and D.G. Albertson. Array comparative genomic hybridization and its applications in cancer. *Nat Genet.*, 37:S11–7, 2005.

[47] D. Pinkel, R. Segraves, D. Sudar, S. Clark, I. Poole, D. Kowbel, C. Collins, W.L. Kuo, C. Chen, Y. Zhai, S.H. Dairkee, B.M. Ljung, J.W. Gray, and D.G. Albertson. High resolution analysis of dna copy number variation using comparative genomic hybridization to microarrays. *Nature Genetics*, 20:207–211, 1998.

[48] J.R. Pollack, C.M. Perou, A.A. Alizadeh, M.B. Eisen, A. Pergamenschikov, C.F. Williams, S.S. Jeffrey, D. Botstein, and P.O. Brown. Genome-wide analysis of dna copy-number changes using cdna microarrays. *Nature Genetics*, 23:41–46, 1999.

[49] J.R. Pollack, T. Sorlie, C.M. Perou, C.A. Rees, S.S. Jeffrey, P.E. Lonning, R. Tibshirani, D. Botstein, A.L. Borresen-Dale, and P.O. Brown. Microarray analysis reveals a major direct role of dna copy number alteration in the transcriptional program of human breast tumors. *PNAS*, 99:12963–12968, 2002.

[50] C. Postma, M.A. Hermsen, J. Coffa, J.P. Baak, J.D. Mueller, E. Mueller, B. Bethke, J.P. Schouten, M. Stolte, and G.A. Meijer. Chromosomal instability in flat adenomas and carcinomas of the colon. *J Pathol*, 205:514–521, 2005.

[51] D.R. Rhodes, J. Yu, K. Shanker, N. Deshpande, R. Varambally, D. Ghosh, T. Barrette, A. Pandey, and A.M. Chinnaiyan. Large-scale meta-analysis of cancer microarray data identifies common transcriptional profiles of neoplastic transformation and progression. *PNAS*, pages 9309–9314, 2004.

[52] J.A. Rice. *Mathematical Statistics and Data Analysis Second Edition*. Duxbury Press, 1995.

[53] A.I. Saeed, V. Sharov, J. White, J. Li, W. Liang, N. Bhagabati, J. Braisted, M. Klapa, T. Currier, M. Thiagarajan, A. Sturn, M. Snuffin, A. Rezantsev, D. Popov, A. Ryltsov, E. Kostukovich, I. Borisovsky, Z. Liu, A. Vinsavich, V. Trush, and J. Quackenbush. Tm4: a free, open-source system for microarray data management and analysis. *Biotechniques*, 34:374–378, 2003.

[54] B. Schölkopf, J. Platt, J. Shawe-Taylor, A. J. Smola, and R. C. Williamson. Estimating the support of a high-dimensional distribution. *Neural Computation*, 13(7):1443–1472, 2001.

[55] B. Schölkopf, R. Williamson, A. Smola, J. Shawe-Taylor, and J. Platt. Support vector method for novelty detection. *Advances in Neural Information Processing Systems*, 12:582–588, 2000.

[56] M.W. Schreurs, M.A. Hermsen, R.I. Klein Geltink, K.B. Scholten, A.A. Brink, E.W. Kueter, M. Tijssen, C.J. Meijer, B. Ylstra, G.A. Meijer, and E. Hooijberg. Genomic stability and functional activity may be lost in telomerase transduced human cd8+ t lymphocytes. *Blood*, 106:2663–2670, 2005.

[57] G. Schwarz. Estimating the dimension of a model. *The Annals of Statistics*, 6(2):461–464, 1978.

[58] J. Sebat, B. Lakshmi, J. Troge, J. Alexander, J. Young, P. Lundin, S. Maner, H. Massa, M. Walker, and M. Chi. Large-scale copy number polymorphism in the human genome. *Science*, 305:525–528, 2004.

[59] A.M. Snijders, J. Fridlyand, D.A. Mans, R. Segraves, A.N. Jain, D. Pinkel, and D.G. Albertson. Shaping of tumor and drug-resistant genomes by instability and selection. *Oncogene*, 22:4370–4379, 2003.

[60] A.M. Snijders, N. Nowak, R. Segraves, S. Blackwood, N. Brown, J. Conroy, G. Hamilton, A.K. Hindle, B. Huey, K. Kimura, S. Law, K. Myambo, J. Palmer, B. Ylstra, J.P. Yue, J.W. Gray, A.N. Jain, D. Pinkel, and D.G. Albertson. Assembly of microarrays for genome-wide measurement of dna copy number by cgh. *Nature Genetics*, 29:263–264, 2001.

[61] A.M. Snijders, M.E. Nowee, J. Fridlyand, J.M. Piek, J.C. Dorsman, A.N. Jain, D. Pinkel, P.J. van Diest, R.H. Verheijen, and D.G. Albertson. Genome-wide-array-based comparative genomic hybridization reveals genetic homogeneity and frequent copy number increases encompassing ccne1 in fallopian tube carcinoma. *Oncogene*, 22:4281–4286, 2003.

[62] A.M. Snijders, B.L. Schmidt, J. Fridlyand, N. Dekker, D. Pinkel, R.C. Jordan, and D.G. Albertson. Rare amplicons implicate frequent deregulation of cell fate specification pathways in oral squamous cell carcinoma. *Oncogene*, 24:4232–4242, 2005.

[63] D. M. J. Tax and R. P. W. Duin. Support vector domain description. *Pattern Recognition Letters*, 20(1113):1191–1199, 1999.

[64] M.A. van de Wiel, S.J. Smeets, R.H. Brakenhoff, and B. Ylstra. Cghmultiarray: exact p-values for multi-array comparative genomic hybridization data. *Bioinformatics*, 21:3193–3194, 2005.

[65] H. van Dekken, P.L. Paris, D.G. Albertson, J.C. Alers, A. Andaya, D. Kowbel, T.H. van der Kwast, D. Pinkel, F.H. Schroder, K.J. Vissers, M.F. Wildhagen, and C. Collins. Evaluation of genetic patterns in different tumor areas of intermediate-grade prostatic adenocarcinomas by high-resolution genomic array analysis. *Genes, Chromosomes and Cancer*, 39:249–256, 2004.

[66] V.N. Vapnik. *Statistical Learning Theory.* John Wiley & Sons, 1998.

[67] P. Wang, Y. Kim, J. Pollack, B. Narasimhan, and R. Tibshirani. A method for calling gains and losses in array cgh data. *Biostatistics*, 6:45–58, 2005.

[68] M.M. Weiss, E.J. Kuipers, C. Postma, A. M. Snijders, I. Siccama, D. Pinkel, J. Westerga, S.G.M. Meuwissen, D. G. Albertson, and G.A. Meijer. Genomic profiling of gastric cancer predicts lymph node status survival. *Oncogene*, 22:1872–1879, 2003.

[69] M.M. Weiss, E.J. Kuipers, C. Postma, A.M. Snijders, M. Stolte, M. Vieth, D. Pinkel, S.G. Meuwissen, D. Albertson, and G.A. Meijer. Genome wide array comparative genomic hybridisation analysis of premalignant lesions of the stomach. *Mol Pathol.*, 56:293–298, 2003.

[70] M.M. Weiss, A.M. Snijders, E.J. Kuipers, B. Ylstra, D. Pinkel, S.G. Meuwissen, P.J. van Diest, D.G. Albertson, and G.A. Meijer. Determination of amplicon boundaries at 20q13.2 in tissue samples of human gastric adenocarcinomas by high-resolution microarray comparative genomic hybridization. *J. Pathol.*, 200:320–326, 2003.

[71] G. Zardo, M.I. Tiirikainen, C. Hong, A. Misra, B.G. Feuerstein, S. Volik, C.C. Collins, K.R. Lamborn, A. Bollen, D. Pinkel, D.G. Albertson, and J.F. Costello. Integrated genomic and epigenomic analyses pinpoint biallelic gene inactivation in tumors. *Nat.Genet.*, 32:453–458, 2002.

[72] X. Zhao, B.A. Weir, T. LaFramboise, M. Lin, R. Beroukhin, L. Garraway, J. Beheshti, J.C. Lee, K. Naoki abd W.G. Richards, D. Sugarbaker, F. Chen, M.A. Rubin, P.A. Janne, L. Girard, J. Minna, D. Christiani, C. Li, W.R. Sellers, and M. Meyerson. Homozygous deletions and chromosome amplifications in human lung carcinomas revealed by single nucleotide polymorphism array analysis. *Cancer Res*, 65:5561–5570, 2005.

Acknowledgments

In this section I would like to thank everyone who contributed to this thesis in any way. Although there are many people I could mention by name here, I will restrict myself to those whose direct contribution was greatest over the entire period in my opinion. They are Aad van der Vaart, Elena Marchiori and Bauke Ylstra. Furthermore, I would like to thank the members of the reading committee for their useful comments.

Samenvatting

Het onderwerp van dit proefschrift is "Machinaal leren voor menselijk kanker onderzoek". Algoritmes die onder machinaal leren vallen, kunnen op veel gebieden van kanker onderzoek worden toegepast. Dit proefschift is ingedeeld naar de centrale informatiestroom die in een menselijke cel plaatsvindt. Deze houdt in dat DNA resulteert in RNA dat op zijn beurt in eiwitten resulteert.

Hoofdstuk 1 is de introductie, hoofdstuk 2 geeft een beknopte introductie in moleculaire biologie en hoofdstuk 3 geeft een korte introductie in de computationele methoden die gebruikt zijn.

Hoofdstuk 4 presenteert een algoritme om ruis te verminderen in de data gegenereerd door de array CGH techniek. Met behulp van array CGH experimenten kunnen chromosomale afwijkingen in cellen van tumor weefsel worden vastgesteld. Chromosomale afwijkingen op specifieke locaties op het genoom kunnen een aanwijzing zijn voor de aanwezigheid van genen die kankervorming onderdrukken of juist versterken. Er zijn echter veel potentiële bronnen van ruis in de experimenten. Om dit probleem aan te pakken hebben wij een "Smoothing" (glad maken) procedure en daarop gebaseerd programma ontwikkeld. De chromosomale afwijkingen die wij hiermee hebben gevonden, zijn vergelijkbaar met die van een expert die de onbewerkte data beoordeelt.

Hoofdstuk 5 beschrijft een methode om array CGH experimenten van verschillende platformen met elkaar te kunnen vergelijken, met een toepassing op een grote data set. Array CGH platformen kunnen verschillen in een aantal opzichten, zoals het aantal lokaties op het genoom dat gemeten wordt en welke lokaties er gemeten worden, maar ook in de ruisverdeling. Het kunnen vergelijken van verschillende platformen stelt ons in staat op eenvoudige wijze veel experimenten uit verschillende studies te kunnen vergelijken. De methode die wij hebben ontwikkeld maakt onder andere gebruik van ons "Smoothing" algoritme. De resultaten suggereren dat onze methode een succesvolle vergelijking van verschillende platformen mogelijk maakt.

Hoofdstuk 6 verschuift de aandacht van het DNA-niveau naar het RNA-niveau. Het hoofdstuk beschrijft een nieuwe clusteringmethode die gebruik maakt van de SVM classificatiemethode. De expressieniveaus van de genen in een cel bepalen hoe een cel zich gedraagt en kunnen dus interessante informatie geven over de tumorcellen. Vanuit een informatica perspectief is het interessant te proberen ideeën uit een succesvolle classificatiemethode toe te passen in

een clusteringalgoritme. Hierbij hebben wij het clusteringprobleem vertaald naar een classificatie probleem met 2 klassen. De resultaten geven aan dat de clustering redelijk robuust is en zich kan meten met een aantal gangbare en goede clusteringalgoritmen.

Hoofdstuk 7 verschuift de aandacht van het RNA-niveau naar het eiwitniveau. Een methode gebaseerd op Support Vector Machines en evolutionaire algoritmen wordt beschreven. Er wordt gebruik gemaakt van data van 2 soorten kanker en van data van gezonde personen. Hiermee worden delen van het SELDI-TOF spectrum achterhaald, die belangrijk zijn om onderscheid te kunnen maken tussen de zieke en gezonde personen. Een serie van experimenten geeft aan dat een relatief kleine verzameling punten uit het SELDI-TOF spectrum veelvuldig terugkeert.

Tenslotte geeft hoofdstuk 8 een beknopte conclusie van dit werk.